Pedro Santos Tiple

Licenciado em Engenharia Informática

Tool for Discovering Sequential Patterns in Financial Markets

Dissertação para obtenção do Grau de
Mestre em Engenharia Informática

Orientador: Nuno Cavalheiro Marques, Professor Auxiliar,
Universidade Nova de Lisboa

	Júri:
Presidente:	João M. S. Lourenço
Arguente:	Luís Cavique
Vogal:	Nuno Cavalheiro Marques

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Novembro, 2014

Tool for Discovering Sequential Patterns in Financial Markets

Acknowledgements

First I would like to thank my thesis advisor, Nuno Cavalheiro Marques, for the given advice and orientation in the process of writing this thesis. For the competences that helped me reach where I am today, I thank all the teachers that I had in the course of my academic life so far.

Special thanks to my family Isabel, Manuel, and João, for their support, patience, and help given through all these years. Especially to my brother for the company and for listening to all my rants.

I would also like to thank my friends Alexandre Pinote, Eládio Oliveira, Filipe Dinis, Filipe Trindade, and Miguel Lagarto for their support, ideas, comments, and motivation.

To Carlos Gomes and João Vasconcelos for going out of their way to give me help and insight on their area of expertise, which was very helpful to improve this thesis.

Abstract

The goal of this thesis is the study of a tool that can help analysts in finding sequential patterns. This tool will have a focus on financial markets. A study will be made on how new and relevant knowledge can be mined from real life information, potentially giving investors, market analysts, and economists new basis to make informed decisions.

The Ramex Forum algorithm will be used as a basis for the tool, due to its ability to find sequential patterns in financial data. So that it further adapts to the needs of the thesis, a study of relevant improvements to the algorithm will be made. Another important aspect of this algorithm is the way that it displays the patterns found, even with good results it is difficult to find relevant patterns among all the studied samples without a proper result visualization component. As such, different combinations of parameterizations and ways to visualize data will be evaluated and their influence in the analysis of those patterns will be discussed.

In order to properly evaluate the utility of this tool, case studies will be performed as a final test. Real information will be used to produce results and those will be evaluated in regards to their accuracy, interest, and relevance.

Keywords: Sequential patterns, ramex forum, market asset relation, associative rules, market dependencies.

RESUMO

O objetivo desta dissertação é o estudo de uma ferramenta que ajude analistas a encontrar padrões sequenciais. Um estudo vai ser feito para averiguar como esta ferramenta permite a descoberta de informação nova e relevante, tendo como foco os mercados financeiros. Nomeadamente pretende-se estudar até que ponto este tipo de método pode possibilitar novas bases para investidores, analistas de mercado, e economistas tomarem decisões informadas.

Devido à sua capacidade para encontrar padrões sequenciais em dados financeiros, o algoritmo Ramex Forum vai ser usado como base para esta ferramenta. Será elaborado um estudo sobre a implementação de diversas melhorias relevantes. A maneira como os padrões encontrados são apresentados é outro aspeto importante deste algoritmo. A utilização de um componente de visualização apropriado, é essencial para encontrar padrões relevantes entre todas as amostras estudadas. Assim, vão ser testadas diferentes combinações de parâmetros e diversas formas de visualizar dados. A influência de cada um destes aspetos na análise de padrões será discutida.

Como teste final, serão realizados diversos casos de estudo para avaliar a utilidade desta ferramenta. Será usada informação real para produzir resultados que serão avaliados relativamente à sua precisão, interesse, e relevância.

Palavras-chave: Padrões sequenciais, ramex forum, relações entre bens de mercado, regras associativas, dependências de mercado.

CONTENTS

List of Figures

List of Tables

LISTINGS

CHAPTER 1

INTRODUCTION

1.1 Motivation

Price variations in the stock market have complex natures and the multitude of factors involved in them are expressed by a large number of variables, in problems where relations emerge. These variables are mostly unknown and it's hard to find how influential each variable is in the price. It would then be useful for an analyst to have a way to facilitate that search. This can be done by taking advantage of computers, with their ability to analyze large volumes of information and to display them in a human readable format. For that an appropriate algorithm is necessary, one that can produce useful results in a short time. The research, planning, and validation of such an algorithm will then be the focus of this thesis.

Due do the large amount of information generated by the stock market, there is access to vast repositories of data that can be combined and evaluated to generate indicators of some market behaviors. This creation of indicators is already done in large scale by traders and technical analysts for a big number of market sectors, leading to an even larger repository of market information. With all the possible combinations of information, it's difficult for a human being to reach specific conclusions, and because of that, financial markets took to using computers in an attempt to aid in their research. One of the paths that can be taken is to use Data Mining techniques to handle the enormous volume of data.

However, if proper precautions are not taken, the amount of data to analyze might be too much even for computers and the generated results might still turn out unreadable for humans. As such, efficient algorithms, that can solve problems in this area while at the same time generating readable results, began being developed. It will then be necessary to either adapt a solution or develop one from scratch based on existing literature.

The objective of this thesis is to create a tool that finds or helps in finding sequential

patterns in financial data, this data has certain characteristics that justify this objective. Various products listed in the stock market represent a wide range of commodities, from food products like milk and cereals, to oil, or company actives. By looking at existing products in the market, we ask the question if it can be concluded with certainty that some stock products exist in an interconnected network of associations and dependencies among themselves. Using the existing data that shows the behavior of the market in the past, correlations between stock products can be found. With the information derivable from these connections, sector professionals can try to infer causal relations in market behavior. Allowing them to advise their clients when to trade with a greater degree of confidence, given that they would have solid basis on which to make their decisions.

Given that there are already some solid basis from which to work on, one algorithm was pre-selected as the basis for this thesis. The Ramex Forum [CM13] algorithm already does most of what is needed, however it hasn't been properly validated in financial markets and needs some small changes to properly adapt to the thesis' theme. An analysis, evaluation, and enhancement of the algorithm will be done so that it provides the required results.

1.2 Objectives

The main objective of this thesis is to design a tool that finds sequential patterns in financial markets, in order to do that several steps must be taken. The process is started by the study of the Ramex Forum (Section 3.3) algorithm, followed by the study of its improvements. Taking into account other existing algorithms and solutions [Cav07a; CC08; Che+10; Pap+09], changes will be made to the Ramex Forum algorithm and its implementation so that the most useful results are obtained in the shortest time possible. This will require several test runs of the algorithm, to evaluate the results obtained and compare them to the different versions developed. The improved algorithm will be integrated in a tool that meets the following criteria:

- Is parameterizable in a way that allows the adaptation of the algorithm to the analyzed market, so that it is flexible and ready to handle distinct uses.
- Takes as input financial data in a standardized format.
- Discovers sequential patterns existent in the inputted data.
- Displays sequential patterns found in an easily readable way that facilitates the discovery of micro-patterns in the results.

While there is already a planned used for this tool, there are some small variations that the tool must accommodate. For example, all three case studies in this thesis do analysis for the daily value of products, still in the future it might be needed to do an analysis by the hour, minute, or even second and the tool should already provide support for these

kinds of different cases. For example, high frequency trading (HFT) requires an analysis that might go as tight as the second scale, which implies the need for a whole different set of parameters than the ones used on a daily frequency. While HFT is outside the scope of this thesis, the tool produced will still allow its study.

Given that the algorithm needs some parametrization, a sensibility analysis must be made regarding the influence of those parameters and its limits on a useful framework. Namely, by considering interdependencies while assuming medium and long term investments in financial products (a period that is considered by some authors as relevant for reducing investment risk [MG10]). As for the display, a good way to visualize sequential patterns must be studied.

Once a functioning prototype is available, the tool will be tested in real world cases. This will help understand how the tool behaves in real world use, while also allowing the evaluation of the results obtained. With the outcome of this study, further improvements on the tool will be done if needed, and possible future work will be proposed.

1.3 Contributions of the Dissertation

The following points have been studied and are proposed as relevant contributions of this dissertation:

- Implementation, improvement, and testing of the Ramex Forum [CM13] algorithm.
- Planning, implementation, and testing of a visualization tool for the results obtained with Ramex Forum.
- Perform case studies with economical datasets obtained from multiple sources (see Section 2.8.1) to evaluate the results given by the developed visualization tool.

1.4 Thesis Organization

This thesis is divided into six chapters, with the first one being the current introduction. The second chapter deals with related work, some algorithms related to the work to be developed are introduced, possible sources of datasets are discussed, and possible choices for foundations as visualization tools are evaluated according to their degree of possible contribution to the thesis.

In the next chapter the Ramex Forum algorithm is fully introduced, analyzed and tested. First a theoretical introduction is done, to fully explain what the uses for the algorithm are and its functionality. Then the implementation of the algorithm is explained along with developments that were deemed necessary for it to have a practical use and be usable in the context of the tool to be created. Finally some tests that validate the correctness and functionality of the algorithm are explained and their result is presented.

The graphical part of the developed tool is presented in the fourth chapter. A small description of each interface window and available functionalities is done along with pictures that show their look and organization.

The fifth chapter presents three case studies that emulate the usage of the tool in real world problems. The first case study focuses on petroleum and is derivatives, the second on the currency market, and the third on two investment funds. For each case study an introduction is done showing why it is relevant, information about the input data is discussed, and finally the results are covered and evaluated.

The sixth and last chapter will make the final conclusions. Furthermore, it will introduce some possible future improvements for the tool and the algorithm, either ideas or problems that arose during the development of this thesis. These ideas either are out of the scope of the thesis or would take more time than available to develop and implement.

CHAPTER 2

RELATED WORK

In association rule learning literature one of the problems that mostly resembles what is going to be done is the market basket problem [AS94]. In that problem the goal is to find, in a database of transactions, a group of items that are frequently bought together or in sequence, thus creating a basket of products that are tightly related. This problem can be adapted to represent the navigation of a user through a website, the items bought by a costumer on a retail or online shop, or as in the case of this thesis, buy and sell transactions in the stock market.

Because of the nature of the data, the amount of it that has to be handled at once can become unbearable. Since individually comparing each transaction with every other is extremely inefficient and time consuming, several algorithms have been developed. This problem already dates back for several years and new iterations of algorithms with improvements in performance and scalability have been released along the years, it is still an ongoing subject.

One of the uses of market basket analysis is recommender systems. These systems recommend the user what products to buy based on his previous transactions and on the item he is currently purchasing. This is seen often in online shops like Amazon or eBay.

This chapter will introduce the roots for the algorithm used in this thesis. Basic data mining and association rule learning concepts will be presented, along with each related algorithm and field of research that was found relevant for this work. First the Apriori algorithm [AS94] will show the groundwork set for most association rule learning, then Similis [Cav07b] will demonstrate that some efficiency and functional improvements can be done over Apriori with good results. However since these two first options don't take into account sequences in time, Interval Based Mining [AF94] knowledge will be included to tackle that problem. Finally, other fields that could provide different approaches, or interesting ideas for improvements to these problems (Process Mining [VDA12] and Music Information Retrieval [Kos+00; Pam+02; RF01]), are also addressed to give an idea of how

else the theme could possibly be addressed.

While the previously referenced algorithms are interesting solutions for specific cases, they aren't a good fit with what needs to be done in this thesis. For each of them their positive and negative points will be discussed, and it will be seen that they all miss something that needs to be addressed. This chapter introduces both related literature and existing knowledge, while also motivating the need to use the Ramex Forum algorithm as it is the one that closest fits the requirements for this dissertation.

2.1 Apriori Algorithm

One of the pillars of association rule learning, the Apriori algorithm [AS94] was developed by Rakesh Agrawal in 1994. This algorithm has served as a basis for the development of other solutions related to the same problems.

Association rules consist of implications, displayed as $X \Rightarrow Y$, that can be interpreted as "when X happens, Y follows" with different degrees of certainty. These rules are inferred from transaction databases of commercial retailers where frequent patterns of item groupings (known as itemsets) are identifiable, however a rule is only generated if it has a certain level of *support*.

Definition 2.1. Support $supp(X)$ of the itemset X takes as value the number of times the itemset appears in the given transaction database.

The bigger the support of a rule, the more relevant it becomes. A minimum threshold is used so that, if the number of times a given association exists in the database is not over a certain value, it will not count as a rule. This minimum threshold is defined as the minimum support *minsup*.

Algorithm 1: Apriori algorithm as seen on [AS94].

Data: Database of transactions
Result: Answer $= \bigcup_k L_k$;
$L_1 = \{\text{large 1–itemsets}\}$;
for $(k = 2; L_{k-1} \neq \emptyset; k{+}{+})$ **do**
  $C_k = \text{apriori–gen}(K_{k-1})$;
  for all transactions $t \in$ Database **do**
    $C_t = \text{subset}(C_k, t)$;
    for all candidates $c \in C_t$ **do**
      c.count++;
    end
    $L_k = \{c \in C_k \mid \text{c.count} \geq \text{minsup}\}$;
  end
end

Algorithm 1 displays the pseudo-code of the Apriori Algorithm. It starts by taking all the transactions in database and generating the 1-itemset L_1. From there the apriori-gen [AS94] function generates from the previous L_{k-1}, the k-itemset and counts the support for each itemset. A new L_k is created from all the candidate itemsets that have a support bigger than *minsup*.

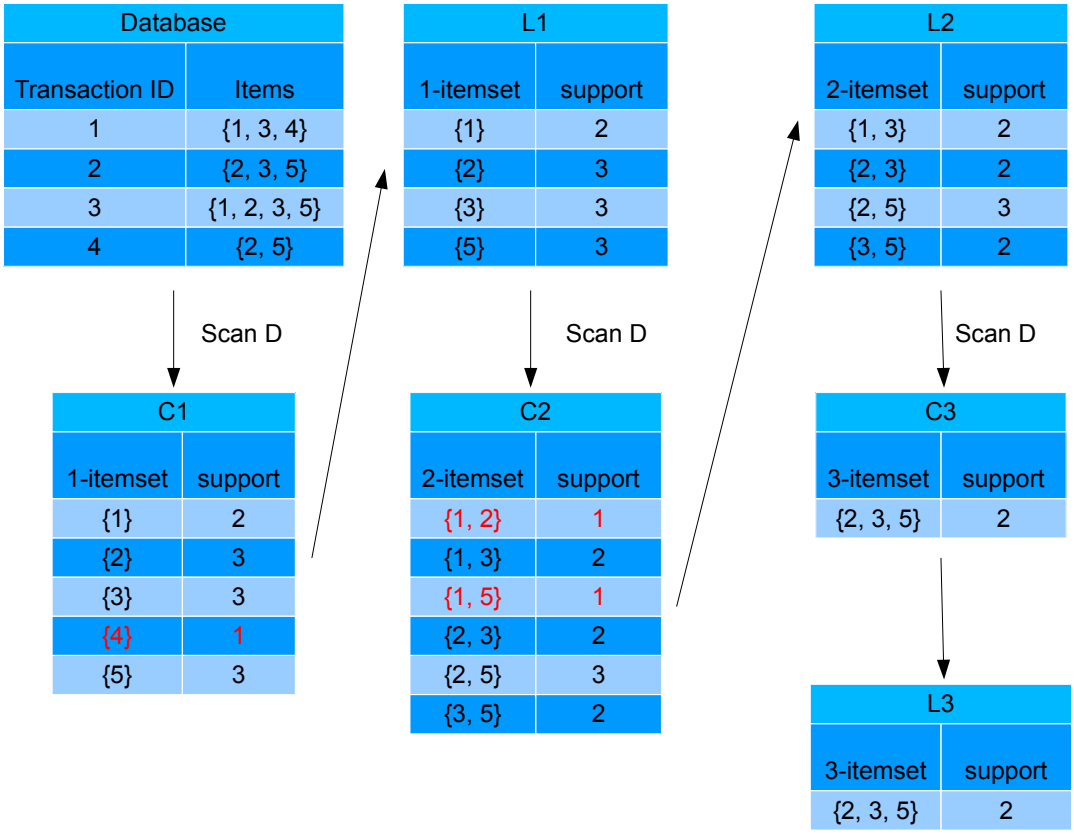


Figure 2.1: Simple example of the Apriori steps based on [AS94].

In **Figure 2.1** there is a very simple example of the algorithm at work, from the database a set of candidates is generated and only the ones that have a support bigger than minsup = 2 are considered. For each L_k a new set of candidates with a higher number of items is generated and a scan over the database (or a structure that partially represents it) is done to count the support. When no more candidates can be generated, the algorithm terminates and the result is all the itemsets of different sizes existent in all of the generated L's.

One of the advantages of the Apriori Algorithm over its predecessors is that it can have more than one item in the precedent of the implication, that means that $X = \{x_1, x_2, ..., x_n\}$ instead of a single item. This leads to richer and more significant rules, however it also

leads to an increase in computational complexity. Which is a problem that resulted in a number of improvements over the algorithm and different approaches to its implementations.

As for disadvantages, the number of items bought in each transaction isn't taken into account, so if an item is frequently bought in bulk, it will be considered as relevant as another item that is bought in the same number of transactions but in a lower quantity. This might lead to the dismissal of relevant rules in some applications of market baskets. Another problem is that in very large datasets the number of rules generated will be hard to analyze due to their large quantity. Since the relevance of the rules is hard to be evaluated by a computer, heavy human involvement is required and this is not optimal.

Due to the previously discussed disadvantages, several different authors built upon Apriori to provide alternatives that solved market basket problems while not incurring in those disadvantages. One common improvements is to define functions that qualify rules, allowing for a way to filter undesired rules from the results. This is further discussed in the next section.

2.2 Association Rule Selection

Before going further, it is important to know that in Association Rule Learning there are three concepts that are frequently used, namely support (already discussed in Apriori), confidence, and lift. These are usually used as minimum threshold to discard generated rules that don't meet the required minimum. They are defined as:

Definition 2.2. The confidence [HK06] of a rule is defined as $conf(X \Rightarrow Y) = \frac{supp(X \cup Y)}{supp(X)}$ and takes as value the percentage of transactions in the database where if X is present in a transaction Y is also in that transaction.

Definition 2.3. The lift of a rule is defined as $lift(X \Rightarrow Y) = \frac{supp(X \cup Y)}{supp(X) \times supp(Y)}$ and "represents a measure of the distance between $P(B|A)$ and $P(B)$ or equivalently, the extent to which A and B are not independent." [McN+08]

These two concepts, more for confidence than lift, are commonly used in the next sections and Ramex Forum redefines confidence to further fit its subject.

There are a number of existing tools that take these concepts and create association rules mining and visualization suites. Most of these tools provide the basic implementation of Apriori and then using minimum support, confidence, and lift restrict the resulting association rules so that they can be more manageable and displayable. Among those tools, are the IBM SPSS Clementine [Sps], a very complete suite of data analysis with the Clementine version being focused on data mining. IBM also has another tool, the "IBM - Intelligent Miner for Data Applications" [Red99]. Finally the R-extension Package arulesViz also provides a number of association rule visualizations that are heavily parameterizable [HC]. However, all these aproaches use the basic principles of Apriori. One way to increase

the association rule legibility is to use explicit algorithms for graph analysis to build even further on top of these concepts to create new approaches, and that is what the Similis Algorithm does.

2.3 Similis Algorithm

The Apriori algorithm led to the development of algorithms based on its idea of association rules. The Similis [Cav07b] algorithm was one of those. This algorithm also creates association rules for market baskets, but takes a different approach.

The Similis algorithm tries do reduce the search space by transforming the transaction database into a weighted undirected graph, where the weights represent the amount of times when the two items are somehow connected in the transactions. Then, within this generated graph there will be occurrences of cliques.

Definition 2.4. Cliques are groups of items that, when separated from all other items not in the group, result in a complete graph.

Thus a clique represents a market basket, a group of items that are all transactioned at once, and by finding the maximum clique the most frequent market basket is found.

Algorithm 2: Similis algorithm as seen on [Cav07b].

Step 1 - Data transformation
Data: support measure, table T
Result: weighted graph G
1) Discard the infrequent 1-itemset using a filter;
2) Generate graph G using the 2-itemset frequency;
Step 2 - Find the maximum-weighted cliques
Data: weighted graph G and size k
Result: weighted clique S of size k
1) Find in G the clique S with k vertices with the maximum weight;

The algorithm is broken down into two steps, first the graph G(Vertices,Edges) is generated from a table T that consists of pairs (transaction, item). Then a search is performed over this graph using the Primal-Tabu meta-heuristic [Cav07b]. This search will find the clique with k vertices and the heaviest weight of all cliques of the same size existent in G. Before the first step, the search space is immediately pruned by removing all items that fall bellow the threshold *minsup*. This means that if an item by itself isn't transactioned a minimum number of times, it will not be considered.

With the data transformation step having a time complexity of $O(N)$ and the search step having a time complexity of $O(N^3)$.

One of the problems with association rules is that a very big number of rules can be generated. This leads to problems with finding relevant rules. And if this is already a

negative aspect in the market basket problem, it's only going to get worse when adapted to economic problems.

Another problem is that both Similis and Apriori don't take into account the sequence of events. Market basket problems only look at items either bought together or by the same person. To find sequential patterns, one of the requirements for the tool, it is important to look at how events relate to each other in time. To solve that problem, Interval-Based mining provides very interesting and relevant contributions.

2.4 Interval-based Mining

Most sequential pattern mining focuses on time-point based patterns, however this approach doesn't apply to some cases. Situations where events persist for periods of time instead of happening instantaneously cannot be properly represented by time-points. In some situations transactions made sequentially over days is not the same if it were over the span of months, it is therefore important to also pay attention to this information.

In order to provide solutions in these instances, Interval-based Mining [Che+10; Pap+09] solutions were developed. These mine frequent temporal patterns as opposed to just frequent patterns. This adds a new layer of complexity and raises some questions such as: "How is the relation between two intervals characterized?", and "How does the relation in time-space affect the relation of two events?".

In related literature the first question is answered with Allen's Relations [AF94] which characterizes the temporal relations of two event intervals A and B according to Figure 2.2.

These 13 relations identify all possible relations between two event intervals, and will be used to identify patterns. It's no longer enough to find sequential patterns, now events will be compared on how frequently they have the same temporal relation to each other. This appears to be straight-forward, on the other hand data collection is not perfect. Some incorrect values might skew a "meets" relation into an "overlaps" one, or any other possible ambiguity shown in Figure 2.2. Not only data collection, but events that do not have completely rigid duration would lead to errors if some way to handle these problems isn't used.

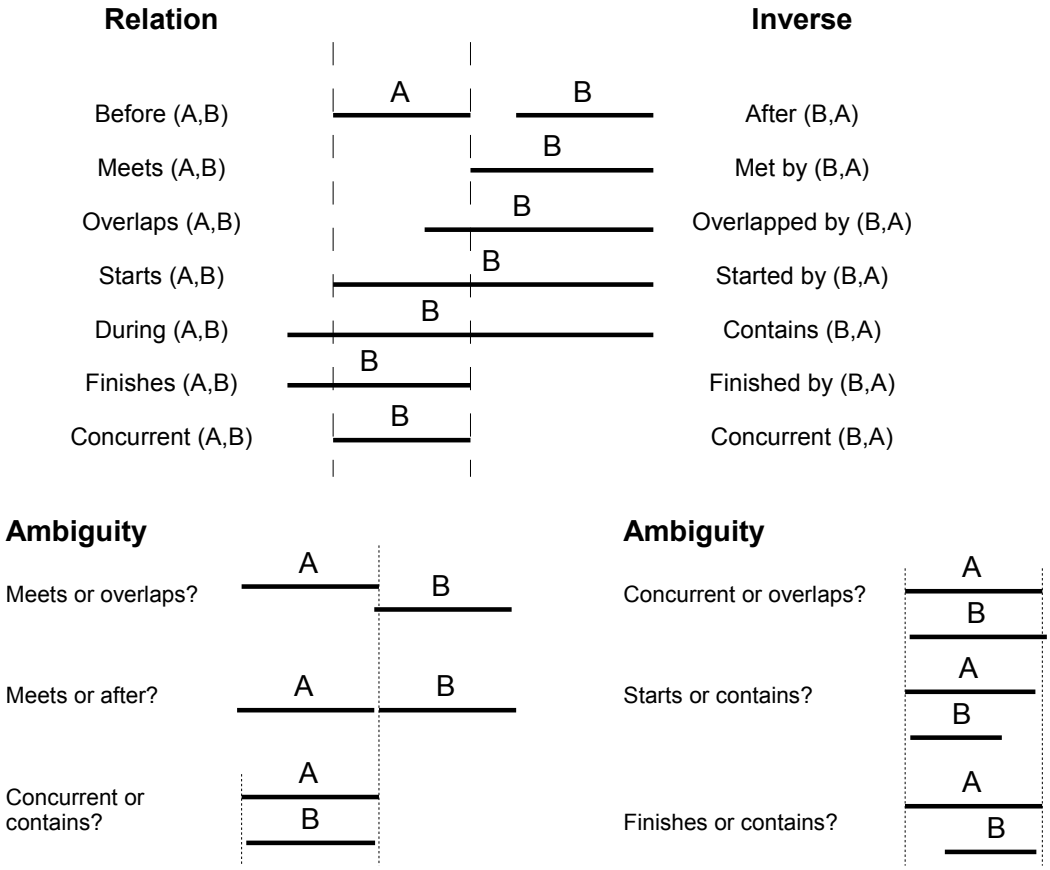


Figure 2.2: Allen's relations between two intervals and possible ambiguities as seen in [AF94].

A new set of relations that makes use of a value ϵ is presented in [Pap+09]. This new parameter is used to allow some slack in relation detection, mitigating the previously referred problems. And in [GQ11] the type of relation is ignored in favor of simply looking at the start time of events and ordering them incrementally which creates a "signature" of the pattern. By using the same ϵ as a "maximal dissimilarity" between signatures, also addresses the problem.

This type of mining is relevant to the thesis' theme because the relation between market assets look only at periods of positive or negative tendency (see Ramex Forum in Chapter 3) and it can be interesting to look at the whole period as one instead of a day-by-day basis like it is being done now. However, this solution provides no flexibility in terms of noise, which is abundant in financial products, as such the results obtained would quickly become buried under irrelevant patterns. This problem in greatly reduced in Ramex Forum because of the way the patterns are searched for and how relations are considered.

2.5 Process Mining

Business processes usually follow specific and semi-rigid models of operation for each activity that is often repeated. These models are either already defined and documented or, because of the way users usually handle processes, the model emerges naturally. Either way they are usually done in sequential steps (even if some steps are done concurrently there is usually a synchronization point that requires all previous steps being complete) that can follow different paths depending on certain conditions.

Using event logs, the authors of [VDA12] had access to many different instances of process execution that could be analyzed in search of sequential patterns. Finding the patterns was done with a simple count of how many times each sequence occurred, however the interesting part is what the found sequences were used for. Three types of process mining are defined as discovery, conformance checking, and enhancement.

The first one, discovery, is analogous to what is done in this thesis, with the input data a behavior model is created that can be used to find interesting and unknown facts, and predict or direct future actions. Analysts [Aal+10] can take information about how employees do their job and given the output:

- generate models of the steps to be taken for each process.
- find the most common processes so that resources can be properly allocated.
- if the input data has the duration of each step, find the steps that are usually holding back the completion of processes.

The second one, conformance checking [RA08], goes a little further and is used to check if the processes are being correctly done according to existing models. Given a model and the event log, frequent patterns are compared to the models to see if, and how, they deviate. This is then used to create a metric to determine the conformance between model and reality. In relation to the thesis, this could be converted to: given the model generated by Ramex Forum(influences, Buy and Sell signals) compare to actual buy and sell orders as a metric of how useful the signals are considered.

The third and final one, enhancement, aims at improving the exiting model with the observed events. As seen with conformance checking, some models might not be adjusted to the actual needs of some processes and need to be updated. This type of mining will find frequent non-conforming sequences and provide them to the analyst so that the model can be improved.

In an economical framing, these approaches could be used to find sequential patterns between products as done in this thesis (discovery), to see how those patterns adjust to future behaviors (conformance checking), and in a more complex setting of cases where funds buy products according to specific policies [IK00], modify and improve the policies so that they provide better results (enhancement).

As already stated, one problem with data mining is how to represent the results in readable ways. There are already several tools that deal with process mining, however each receives input data and displays the output in their own way. The ProM Framework [Don+05] tries to deal with this problem by offering a standardized input and output along with plug-in capabilities. This way different algorithms can be used in the same platform, with the same data, and producing structurally equal outputs so that they can easily be compared and evaluated. This is a good motivation for building the tool in an open way so that it is easy to use in the future by others with different types of data.

While very interesting, process mining works under the assumption that there is already a defined model, which isn't the reality for financial markets. Because the discovery step is already what Ramex Forum does for financial data, the algorithms discussed here would be more useful as a post validation between the results obtained with Ramex Forum and the market behavior.

2.6 Music Information Retrieval

One subject that gained some traction in data mining was the analysis of songs and music. From cataloging genres and artists [RF01] to finding similar songs based on specific song characteristics [Pam+02], or even searching for a song based on small clips of audio [Kos+00], the possibilities for data mining within this subject are extensive. Research that fits in this theme is usually grouped in the data mining sub-genre of music information retrieval (MRI) and there have been a number of different approaches to this subject.

Audio can be characterized by several things including loudness, pitch, tone, amplitude (continuous data), or by notes (categorical data). These characteristics can all be represented graphically (see Figure 2.3), showing the way their value changes along a song sometimes could be easily mistaken for the way a stock price changes. Furthermore, economical data also has more than one characteristic that can, and should, be used to find patterns such as different price categories, volume of sales, or expected yields. It might then be possible to use MRI techniques for pattern and similarity searches in economical data.

In "A Survey Of Music Information Retrieval Systems" [Typ+05], as the title states, the author makes a survey of music information retrieval systems existent at the time (2005) and provides a couple of methods that could be adapted for economical data. Some of the presented systems are not suitable for sequential pattern finding, because they try to find an exact match between characteristics of two audio clips that are a subset of the whole audio clip, they would almost never find any matches since no two products behave exactly the same.

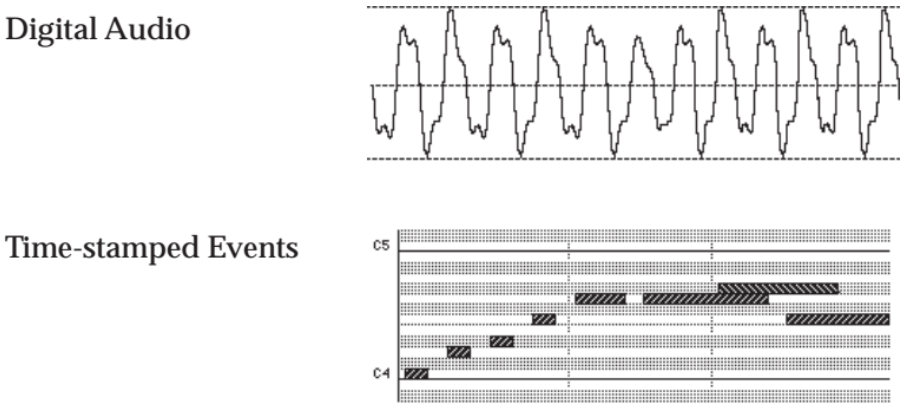


Figure 2.3: Basic representations of music (schematic) [BC02]

The interest is then focused on systems that try to find similar patterns between clips [Kur+02], these usually exist as a solution to noise problems. One of the uses of MRI systems is to identify songs from recordings done with low quality recording devices [WF03], usually in non-optimal sound situations, that lead to a degradation of sound quality and an introduction of external noises. In the article these systems are categorized as "distance measures" in three methods: "string-based methods for monophonic melodies", "set-based methods for polyphonic music", and "probabilistic matching", each tackling the problems in their own way.

Financial products can be looked at as a stabler core (actual product value) that suffers frequent changes caused by investor reactions (noise). By applying MRI systems that deal with these situations in audio terms to financial products, long term patterns could possibly be found.

Still, extensive tests are needed before these solutions can be adapted to finding sequential patterns. In financial markets the ratio between price and moving averages has been found to provide interesting enough signals [MG10]. This way, Ramex Forum is a better choice because it uses moving average based signals (Chapter 3) while MRI based systems remain interesting for future work.

2.7 Sequential Pattern Mining using a Bitmap Representation

The authors of [Ayr+02] propose a new approach to solve the problem and increase performance by using Bitmaps to represent data. The algorithm is named SPAM (Sequential PAttern Mining) and it is based on the Apriori algorithm, some changes were made in order to improve performance. Such as introducing the pruning of some nodes, which reduces the search space considerably and the contribution of the paper, the usage of a bitmap. The bitmap is associated with each item in the itemset. For each transaction if the item is present in that transaction, the entry on the bitmap that corresponds to that

item has a value of 1. After some transformations to the bitmaps, namely the pruning, the sequential pattern finding begins and outputs are generated.

In the paper it is stated that this algorithm, when compared to SPADE [Zak01] and PrefixSpan [Pei+01], can find patterns an order of magnitude faster in large datasets and that in small datasets the difference is not so significant. One of the problems indicated is that because of how bitmaps are used, there is some space inefficiency. However this can somewhat mitigated by compressing the bitmaps. This solution may become necessary if the temporal complexities of the Ramex Forum start to become impractical.

2.8 Existing Datasets

Given that the objective is to find sequential patterns in financial markets, there is a need for large volumes of the data generated by them. Fortunately, since these markets are mostly public and used by a large variety of people and institutions, there are several sources on the internet that supply daily information about the status and changes of the markets. The type of information supplied, depth, and frequency varies from source to source, but there are several that are mostly regarded as accurate, reliable, and trustworthy. In this section several sources will be analyzed and discussed in order to find the ones that better fit the requirements.

In order to evaluate the behavior and output of the algorithms there is a need for sets of data data meet specific requirements, such as continuity, correctness, and type of data. For the:

- Type of Data – the best option would be binary categorical data, however, this type of data does not occur in many interesting situations. Because of that, the requirement is set in types of data that can be translated into binary categories without losing relevant information.

- Correctness – data must be correct, accurate, and reliable or else results might turn out inconclusive or downright incorrect. It is then necessary to use reliable and accredited sources, so that there is a certainty that what is seen as results isn't negatively influenced by the base data.

- Continuity – sequential patterns necessarily involve information that somehow is defined as continual, be it a date, a value that keeps the order of events, or even precedence/sequence indicators. If this information isn't all present and continuous, these temporal or sequential "holes" will produce results that might not represent the actual behavior of the events that are recorded in the data.

2.8.1 Dataset sources

Not only is financial data required, there is also a need to know of relevant events or major changes in the involved economies so that outlier patterns that might show in the

evaluation of the final work output can be understood and justified.

There are several repositories of information on the internet that contain relevant data, several were picked based on the information they provide and the relevant ones were analyzed.

2.8.1.1 Used sources:

The following information sources were used:

Yahoo Finance [YF]: this is a multi-purposed hub for financial information, providing stock quotes, financial news, exchange rates and important global indices. It is therefore an important source of up-to-date and past information of economical markets. One of the advantages of this source is that most of the data can be imported into a spreadsheet format, facilitating the formatting of the information into the required form.

Google Finance [GF]: One of the many facets of Google, this time focusing on finance. Great source for related news and was mostly used to obtain important events in the financial world or historical prices for products not found in other sources.

U.S Energy Information Administration [Eia]: this source is an official, congress funded, collector and disseminator of energy information mostly related to the United States of America. It was used to obtain the prices of oil derivatives used in the third case study (Section 5.3). It was the only reliable looking source found for that kind of information since these kinds of prices present in normal stock or fund exchanges.

Federal Reserve Bank of St. Louis [Stl]: an american government related organization that supervises financial institutions while providing some information relevant for its field of operations. It was a useful source of historical data for currency exchange rates related to the US dollar, since the other sources used didn't provide historical prices for currencies.

Ariva.de [AR]: is a german website that provides a lot of information about financial products, it has the advantage that it provides historical price data for several exchanges and usually both in Euro and US Dollar. Fund prices, for the most part, aren't covered by Google Finance or Yahoo Finance so ariva.de was the primary source for fund princes in the second (Section 5.2) and third (Section 5.3) case studies.

2.8.1.2 Sources not used :

Other sources of information were initially considered, however there was no need to use them. For reference here are some other useful sources of financial related information:

Bloomberg [Blo]: A great source for everything finance related and one of the most complete sources for economical information. However its strong point is market indicators and processed data while for this thesis the focus is on using accountable information. In the future when it might become important to do a further analysis of studied components with this type of information, Bloomberg might be considered as a source.

World Bank Group [Wor]: great source for development and status indicators of countries around the world. Provides data in a wide range of categories that can help understand the changes in the financial world, or in reverse, that can show how the financial status of an economy can change the life of the people.

CIA World Fact Book [Cia]: the CIA provides some information they have collected for every country in the world. While not very extensive, it provides some important indicators of the country's general economic status in an accessible way.

Pordata [Por]: a database of almost everything related to Portugal, from health to education, culture and economics this database provides a lot of information specific to this one country. While having a primary focus on Portugal, it also provides the same information for other European countries in the same format, facilitating comparisons.

2.9 Existing Visualization Tools

In Section 2.2 a couple of existing tools for association rule visualization were already presented, however these tools are used for association rules, not sequential patterns. As such some other way to display the sequential patterns must be devised.

Existing visualization tools, such applications or libraries, are discussed in this section and their advantages/disadvantages in relation to the thesis will be analyzed. First a couple of well known tools for association rule visualization are presented, and then other alternatives geared towards graphical displays are discussed.

Five different applications and libraries were picked due to their ability to atleast display graphs, allow data input from external file-types and some way to integrate or customize into the proposed visualization tool for the thesis.

2.9.1 Graphviz

Graphviz [Gra] is an open source visualization application from AT&T that focuses on graph drawing, providing several different choices in the type of graph to draw. However it lacks in terms of customizability of the graphs and the GUI is poor, because of that some external projects and extensions have surfaced in an attempt to give more functionality to this application.

For positive aspects of Graphviz, it is quick, simple, and easy to use for basic graphs, it also has third party interfaces with several programming languages. The most relevant disadvantage is its static and non-interactive graphs.

2.9.2 Gephi

Another open source application specialized in graphs that provides the same functionalities has Graphviz and much more. Gephi [Gep] is described as "a tool for people that have to explore and understand graphs" and as such it has many functionalities that help with the clear visualization of complex graphs.

For positive aspects of Gephi, it has an appellative layout, many relevant functionalities such as dynamic filtering, node coloring and grouping, and data selection by time-line. Development of plug-ins allows application customization which makes it very interesting in case more interactivity is needed, mostly for future work. As for disadvantages, it is still in beta (0.8.2) and has a complex API that would take too much time of the thesis to properly use.

2.9.3 D3 javascript

Set apart from the other tools so far, D3 [D3j] isn't an application but a JavaScript library and as such it is meant to be used on web-pages. Not focused only on graphs, this library allows many different ways to display information and is the most complete so far in regards of display options. Being HTML oriented might be a big benefit for the thesis, since it allows it to be system independent and adds great mobility.

For positive aspects of D3 javascript, it has variety in display choices of data, with customizable appearance of graphs via CSS in Web-pages that provide a platform independent results. As for negative aspects, it is hard to integrate with other programming languages, such as the used Java and Octave mostly because it relies on web-browsers. Who in turn aren't very fast at displaying large information and quickly consume a lot of system resources.

CHAPTER 3

RAMEX FORUM

In this chapter the Ramex algorithm [Cav07a] and its incremental improvements are explained. First an introduction of the algorithms and their behavior, as it is presented in their original literature, is done to situate the reader. After that the improvement, implementation, validation, and discussion of the extended algorithm is done over a series of sections namely, “Implementation”, “Space and Temporal Complexity”, “Algorithm Validation” and “Simulator”.

This algorithm was chosen for its capacity to summarize information in a way that is visually interpretable and that facilitates the discovery of micro-patterns, along with its adaptability to the goal of the thesis and not generating large amounts of association rules. The lack of association rules and the polytree representation allow an easier to read and study output, which is a great basis for the tool do be developed. All these points will show how Ramex Forum fits better to the requirements of this thesis when compared to the existing solutions presented in the previous chapter.

3.1 Ramex

The Ramex algorithm [Cav07a] is a sequential pattern discovery algorithm, and it is the basis upon which the Ramex Forum [CM13] algorithm stands. A sequential pattern is a sequence of events that occur frequently in a certain certain dataset, either relating to the behavior of a single entity or as events between two or more entities. It must be noted that Ramex and Ramex Forum are two algorithms that shouldn’t be confused.

Algorithm 3: Ramex algorithm as seen on [Cav07a].

Data: a database
Result: tree of sequences
1) Network Transformation
1.1) Sort Data;
1.2) Create new attribute: next-item;
1.3) Build a state transition network;
2) Find highly probable branch sequence
2.1) Condensation process;
2.2) Unraveling process;

Algorithm 3 first creates a weighted directed graph where each node is an item, the edges represent a precedence between items and the weights are the number of times this precedence occurs. After the network has been built, a search is done over it to find a tree that represents the most probable sequences of items. The final output of the algorithm is a tree structure that represents a sequence of events that starts from a root and branches to different parallel sequences.

Much like Similis (Section 2.3), Ramex has two phases, one that transforms the data and another that does the search for the best solution. In the transformation step the database is converted into a network – a graph with a source/root and a sink – with cycles so that all sequences can be represented.

In the second stage the Maximum Weight Rooted Branching Algorithm is applied over the generated network, this will remove cycles while keeping all nodes connected with the heaviest edges, thus generating the expected tree structure. This is done in two steps, first by condensing the previously generated graph and after that by unraveling the new condensed one.

The condensation process condenses cycles by removing all the nodes contained in cycles that aren't exclusively necessary to reach another node. This will set the graph up for the unraveling process, which takes this acyclic graph and unravels nodes in a way that doesn't generate cycles while using the heaviest edges.

The result is a polytree that represents the sequences found. A polytree is a "directed acyclic graph with the property that ignoring the directions on edges yields a graph with no undirected cycles" [Das99].

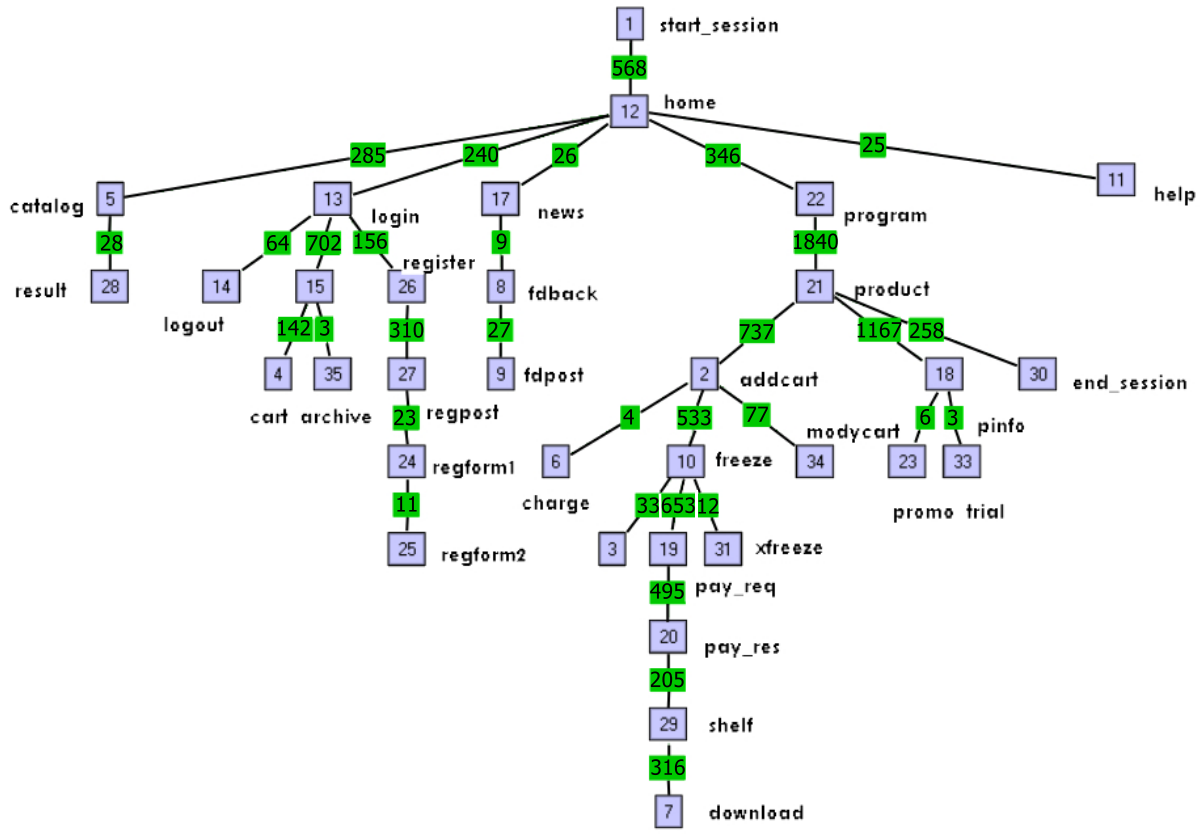


Figure 3.1: Most common user navigations within a e-commerce website, result obtained with the Ramex algorithm [Cav07a].

While not a primary focus for this dissertation, in computer science it's always important to take into account how computationally heavy an algorithm is. For Ramex, in terms of complexity, the first step requires the sorting of all entries in the database. This is a very basic action that database management systems already provide, however, in the worst case scenario it will incur in a $O(N^2)$ time complexity with N being the number of entries in the database. The building of the state transition network will require a new pass over all N entries. In the second step with the Maximum Weight Rooted Branching Algorithm there is a worst case scenario complexity of $\Theta(N^2)$, N being the number of vertexes. Thus, the algorithm has a time complexity of $O(N^2 + N + N^2)$, which is simplified to $O(N^2)$. It can be then concluded that in terms of time complexity, this algorithm is viable for medium-large databases.

3.2 Forward and Back-and-Forward Heuristics

The Forward and Back-and-Forward Heuristics [CC08] were introduced in an attempt to reduce the time complexity of the Ramex algorithm. While its complexity was viable, the scalability was not ideal, for cases with very large datasets the quadratical complexity can become problematic. Since there is no polynomial algorithm to find the solution for directed graphs[CC08], the Forward and Back-and-Forward heuristics were developed and used.

The algorithm itself was kept as seen in **Algorithm 3** with one change, in step 2 instead of using the Maximum Weight Rooted Branching Algorithm one of the heuristics is used.

Both heuristics were based on the Prim algorithm and they are quite similar to each other, the difference between them being the type of tree they output and the existence or not of a starting vertex.

Algorithm 4: Forward heuristic as seen on [CC08].

```
Data: a network G
Result: tree of sequences S
Initialize S;
for each vertex of G do
    for each edge of G do
        x = arg_max(vertex weighted ahead, not visited in G and connected to S);
    end
    Update S with x;
end
```

Algorithm 5: Back-and-Forward heuristic as seen on [CC08].

```
Data: a network G
Result: tree of sequences S
Initialize S;
for each vertex of G do
    for each edge of G do
        x = arg_max(vertex weighted ahead, not visited in G and connected to S;
        vertex weighted before, not visited in G and connected to S);
    end
    Update S with x;
end
```

The Forward heuristic outputs a simple tree and is used in cases where there is a given starting point. For example in website navigation, most users will start their interaction in the home page and navigate the website from there. With the Back-and-Forward heuristic the output is a polytree and doesn't require a starting point.

As for the heuristic's complexity, based on Algorithms 4 and 5 it is observable that the cycles run V*E times, V being the number of vertices of G and E the number of edges of G. With that, the time complexity can be taken as $O(V \times E)$, a polynomial complexity that gives better time performance than the one Ramex without heuristics could achieve.

3.3 Ramex Forum

In order to use Ramex directly in the financial market, a new improvement was proposed as the Ramex Forum [CM13], the objective of this new algorithm is to find relations between different financial market assets.

New challenges arose because of the kind of data that is used, it's no longer about finding sequences in databases of well defined transactions. Now large datasets of financial information have to be converted so that they have a categorical representation. Having this representation is important because, due to the nature of the data there will be a very big amount of diverse values, this diversity makes it very hard to find relations between data and their influences.

The type of data that this version is aimed at is stock market asset prices and their trading volumes, in order to convert these values into categorical data first the categories were defined as: the buying tendency B and the selling tendency S.

For this algorithm at any given point in time an asset can be in one of three states, tendency to increase (B), decrease (S), or unstable. The calculation of the tendency is based in either the total amount of money traded by that asset in a day or its price(I_{value}), and the moving average(MA) of the chosen base value.

In order to have relevant and reliable indicators while calculating the tendency, several steps and calculations must be done. First an Asset Signal Indicator [CM13] ρ_I will be calculated:

$$\rho_I(t,n) = \frac{I_{value}(t)}{MA(t,n)}$$

With the moving average, calculated in the interval $[t-n, t]$:

$$MA(t,n) = \frac{\sum_{i=t-n}^{t} I_{value}(i)}{n}$$

The authors of [CM13] proposed that each category has a counter associated to it, and that these counters will change value according to the following rules:

$$Counter_B = Counter_B + 1 : \rho_I(t,n) \geq 1.05$$

$$Counter_S = Counter_S + 1 : \rho_I(t,n) \leq 0.95$$

$$Counter_B = 0, Counter_S = 0 : 0.95 < \rho_I(t,n) < 1.05$$

These rules mean that if the Asset Signal Indicator is at least slightly above the moving average (positive) then that day there was a tendency to buy, and if it is at least slightly below the moving average (negative) then there was a tendency to sell.

However if the distance to the moving average isn't significant enough, the counters are reset. This is done because if the signal indicator moves close enough to the moving average then there is a significant probability that the tendency is going to change and if the counters are reset each time this happens, they will effectively take the value of how many consecutive days an asset spent in a buy or sell tendency.

There is room for improvement here as some situations might benefit from a non fixed interval. For example, a new value ϵ like the one seen before in Section 2.4 could be used to parameterize the interval into $]1-\epsilon; 1+\epsilon[$.

With this information calculated there is enough concise data and it's now possible to try to deduce the relations and influences between assets.

Definition 3.1. The parameter δ is the maximum trading period length, in days, where a check for relations between two assets is made.

With the function *InfluenceCue(A,B, δ)* a check is done for an influence:

$$A \Rightarrow B : 1 \leq |Count_A| - |Count_B| \leq \delta$$

Where

$$Count_X = Counter_B(X) - Counter_S(X)$$

This means that if the number of days that the two assets share the same tendency is more than zero, and lower or equal to the specified trading period δ, then there is an influence from A to B. As such δ is effectively the maximum allowed distance between the counter start of each product. Because of the way the counters are incremented and reset, and the way $Count_X$ is calculated, it's ensured that when an asset changes tendency, its relation to other assets is broken.

By doing this for each asset in a time period and considering each of these rules, a graph can be generated where each implication is an edge. Furthermore, when multiple edges (one for each day) occur between the same two assets, if the weight of that edge is set to be the number of times the edge exists, a weighted graph is generated like the ones used in Ramex. The Back-and-Forward heuristic is then used to find the most probable polytree sequence of items.

From now on an edge between two nodes will be referred to as an "influence". As such, any time an edge is referenced it can be with that name, even if the edges' weight is very low and doesn't actually represent a solid case where one node is always influenced by another. While these "influences" might be coincidence and not causal, in this context they will be named as such.

3.3.1 Support and Confidence

In association rule learning the concepts of support, minimum support, and confidence are very important, as seen with Apriori (Section 2.1). While Ramex Forum doesn't produce results in the common association rule format, the influences can still be considered as item sets of size 2 that are represented as $I = \{i_1, i_2\}$ where I is an influence from i_1 to i_2.

By making a relation between the concepts in association rule learning and how Ramex Forum detects events, two new definitions are set, support and confidence, for Ramex Forum.

Definition 3.2. The support $supp(X \Rightarrow Y)$ of the influence $X \Rightarrow Y$ is the percentage of moments in the whole studied period where an event was detected for that influence.

Support indicates how prolific events are, in the studied period, for an influence. A high support will then mean that the influence happens a lot in the analyzed period, while a low support means that there are few occurrences in relation to the size of the studied period. For example, if in 100 moments there are 50 detected events for the influence $X \Rightarrow Y$ then $supp(X \Rightarrow Y)$ will have a value of 50%.

Definition 3.3. The confidence $conf(X \Rightarrow Y)$ of the influence $X \Rightarrow Y$ is the percentage of detected events in the moments where events are expected for X.

An event is expected whenever the influencing product X is above/below the upper/lower threshold, because that is the pre-condition for an event to be detected when Y changes state to follow the same behavior. Confidence will then give the probability of Y being in the same state as X, with at least one less count in consecutive moments in that state, and no more than δ days in difference.

As an example, in a total of 100 moments where both X and Y start with values between thresholds, X rises above the upper threshold in the 10th and stays there until the 20th (10 moments total of expected events). Then Y follows on the 15th and the algorithm detects 5 events (15th until the 20th where X falls below the threshold), $conf(X \Rightarrow Y)$ will then have a value of 50% because out of 10 expected events 5 were detected.

These measures of significance and interest are usually used to set minimum thresholds, *minsup* being the most common for association rules. However for Ramex Forum these will be more useful to simplify the visualization of influences, as seen in Section 3.4.5.

3.4 Implementation

The source code for an implementation of the Ramex Forum algorithm in C was provided by the authors of [CC08], however this implementation wasn't found suitable. There are other languages that provide packages with easier integration with GUI and database access. The choice was then made to use Java and the algorithm was reimplemented in that platform from the ground up. Java was chosen for its portability and ease of development, which allowed a quick implementation and adaptation of the algorithm as well as compatibility with most operating systems.

Several improvements where done over the provided source code, some of those were done in terms of both computational and memory efficiency while others provided more utility to the algorithm. These improvements include integration with a database, parameterization, and output choices. Each one of these will be discussed next.

3.4.1 Class structure

The RamexForum package contains 4 classes: RamexForum, Graph, Node and Edge. The main computational load is done on the RamexForum class, it mainly provides two functions, one for each step of the Ramex Forum algorithm. In each step an instance of Graph is created, the first step - loadGraph() - creates the influence graph from the input information and the second one - prim_WB() - simplifies the graph into a tree, both of which can be represented with the Graph class.

Instances of Graph represent a collection of nodes and edges, the class provides functions to operate over them such as add, delete and modify. Each node and edge existent in an instance of Graph is represented by an instance of Node or Edge, this way each Node keeps the information about the nodes of the graph, each Edge keeps both nodes related to that Edge and each Graph can easily access all the information that represents the graph or tree.

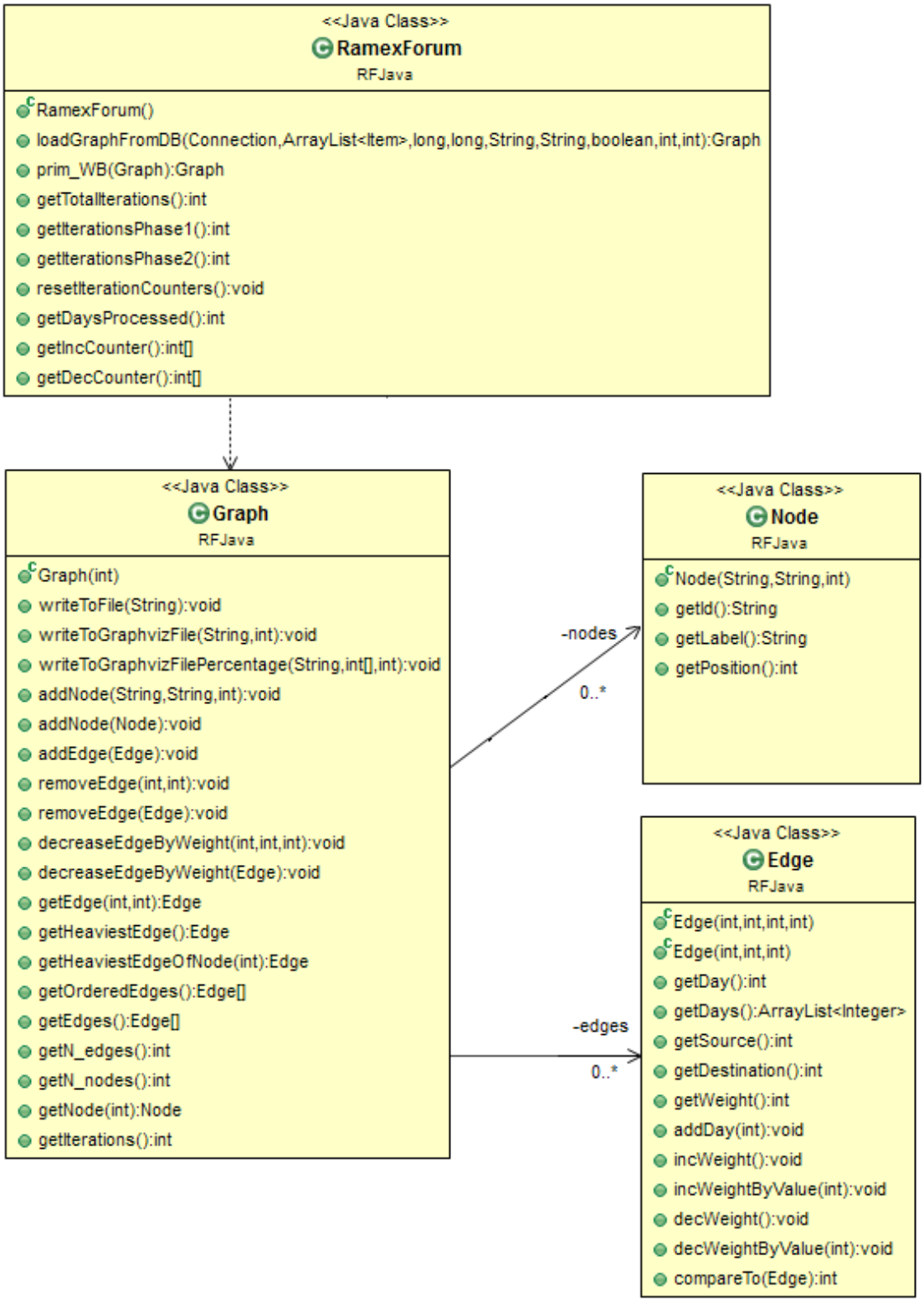


Figure 3.2: Ramex Forum Class Diagram

3.4.1.1 RamexForum.java

To load the initial graph from a database, a connection to the database must be provided, along with: a string that identifies all products that are to be processed, the interval of dates to process, a value type that identifies what values will be used, and the parameterization of all available parameters in the algorithm. This will create an instance of Graph that represents a graph with all the connections between items generated in the first step of the Ramex Forum algorithm as seen in Section 3.3.

The graph is built according to the following algorithm:

Algorithm 6: Graph building with data from a Database

```
Data: Database Connection DBC; Time Interval TI;
Delta δ; Influence Type IT;
List of Items to check IL;
Result: A Graph G of influences;
Initialize G;
Get available moments in TI from DBC;
for each moment M available do
    for each item I of IL do
        Get value of I in the moment M from DBC and increase or decrease the value
        of counter[I] according to the rules seen in Section 3.3.
    end
    for each item I_1 of IL do
        for each other item I_2 of IL do
            Add an edge to G or increase an existing edge's weight if counter[I_1]
            relates with counter[I_2] according to IT and δ.
        end
    end
end
```

The generated graph can then be passed to the prim_WB function so that a tree can be created from the provided Graph, the function implements the Back-and-Forward heuristic, seen in Section 3.2.

The heuristic is implemented in the following way: from the input graph the heaviest edge is added to the output tree as a starting point, then for each node, the list of edges is iterated and the heaviest edge that doesn't cause a cycle and has one vertex on the current output tree is added to the result.

In order to increase the efficiency of the algorithm two steps were introduced: each time a cycle is detected the edge that would cause the cycle is removed from the list of edges; and when an edge is added to the output tree the same edge is removed from the list of edges as well. The remove operation has a constant time complexity and by removing the edge from the list, the next iteration over the nodes will need to search for the heaviest edge in a smaller pool of edges, reducing the number of iterations processed. While not

a big improvement, for the purposes of interactivity in the tool, any time reduced in processing time is less time that the user needs to wait.

3.4.1.2 Graph.java

The Graph class is used to represent graphs and trees used in the Ramex Forum algorithm. A Graph consists of nodes and edges, and as such provides functions to operate over those components. Operations such as add, delete and modify(increase or decrease the weight of edges) are available and necessary for Ramex Forum.

It's also possible to get the heaviest edge, the heaviest edge where one of its nodes is a specific node, and get a list of the edges contained in Graph, ordered or not.

Methods to write a textual representation of the Graph are also available, one to write the graph in a simple text file, one to write in the Graphviz(Section 2.9.1) file format and another one where the weight of the edges is output as percentages instead of simple integers.

For debugging and evaluation purposes, this class keeps a count of cycle iterations it makes.

3.4.1.3 Edge.java and Node.java

Edge and Node are very simple classes and their use is to store information in a more organized format.

An edge keeps its source and destination, its weight and the days(moments) when the edge's weight was increased. These days will be used by the tool to draw charts and each day will be used to represent an event in the chart.

A node simply keeps its ID, label and position within the Graph it is contained in.

3.4.2 Database integration

In order to provide the algorithm with more flexibility in terms of input, it was modified so that the values were taken directly from a database. This modification was not done in the general definition of the algorithm but in its implementation. A database structure was drawn to keep the information in an easily accessible format, this structure can be seen in figure 3.3 and is discussed in Subsection 3.4.2.1.

The database was drawn in a generic way so that different types of values can be kept without many restrictions, this way whenever a new type of value or product needs to be studied it can be quickly inserted into the database and analyzed.

Data to be inserted into the database is collected from several sources, processed using Octave scripts and saved in Octave's text file format. In order to upload the values to the database, a small application was created to parse the text files generated and upload their contents into the database.

3.4.2.1 Database Structure

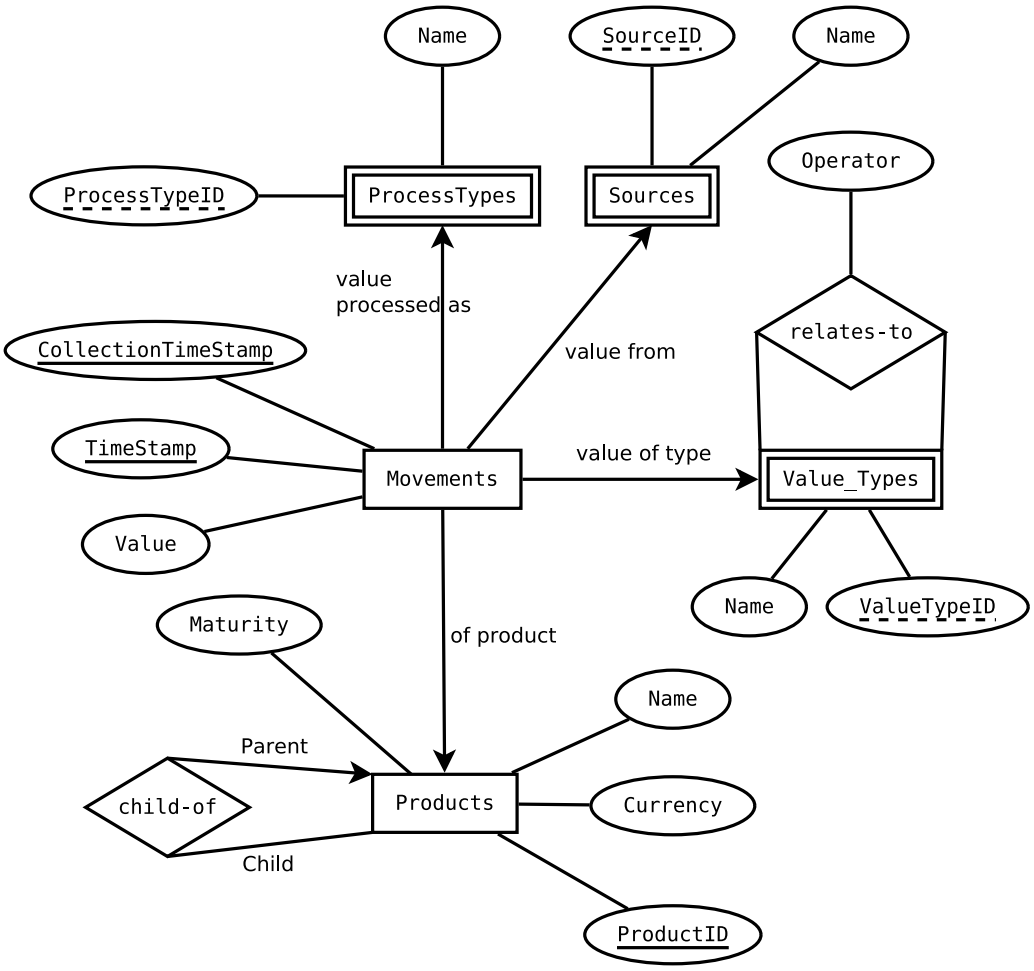


Figure 3.3: Simple database Entity-Relations Diagram using the notation of [Sil+05].

In the entity-relations (ER) diagram seen in Figure 3.3 the table structure of the database can be seen in the usual entity–relationship model: tables (entities) are represented by rectangles; fields are represented by ellipses, if the field is part of the primary key its name will have a full underline, if it is an external key its name will have a dashed underline; many to one relations are represented by directed lines between entities; relation types are defined with diamond shaped nodes; rectangles with double borders are weak entities, these entities share a field (the external key) with another table.

The Movements table contains values related to products and moments in time. To further describe a value other fields are used, such as source, collection time, what the value represents and how it was transformed. The purpose of each field is the following:

- Source: Identifies where the value was obtained from. Since several sources are used, there will be more than one value for each moment in time, this means that a source identifier is required to uniquely identify a value.

- Collection Time (CollectionTimeStamp): It's important to know when a value was collected so that an history of changes to a value in a certain moment can be kept.

- What the value represents (ValueType): In order to provide more flexibility and future proofing to the structure, this field identifies what the value represents. Be it either the highest or lowest bidding price on a given asset or the percentage of unemployment in a country. This way the type of value used isn't restricted to one type and different types of value are clearly identified.

- Transformation (ProcessType): Some values might go through some type of transformation such as normalization, sums, or averaging before they are useful. With this field, in a given moment in time for the same product, it's possible to have several different processed values.

Each measurement is related to a product and each product is kept in the Products table. A product is defined by its name, the currency it uses and its maturity. The last field is an optional for types of products that might have an "expiration date". Products can be related between themselves in a parent/child type of relation, a product can have one parent and many children. This is used to chain relations between products, for example a market index can be the parent of all the products contained in that index.

One value type can relate to another using an operator, for example the normalized closing value of a stock can be related to the 40 day moving average of that value using the division operator. This is used by the tool to have an accessible definition on how values relate to each other without the need for user input. Otherwise the user would have to pick a value-type to study, another one to use as the baseline for the threshold limits (usually a type of average) and the operator. Given that in most cases the relation is value-type to the moving average of the value by division, it was simpler to not allow user input and this way a parameter that is rarely changed can be kept in the database. However the database structure accommodates the ability to implement user input in the future by changing the entries.

3.4.2.2 Future Database Developments

While not necessary for the tool as it currently is, two improvements were thought of that could offer more functionality.

First, allowing the δ in the algorithm to be defined by entries in the database. This way instead of having a fixed δ for the whole interval processed, each timestamp can have a value of δ associated with it. This way different moments have varied weights in creating implications, for example, if in one day there is a major localized event that heavily influences the system behavior only for a short period of time, it might be useful to set a small δ for the time interval of that event. This way only faster relations will be detected in that period while not affecting the rest of the studied time frame.

And second, because the behavior of markets is highly unstable in several ways, one of them being in volume of trades. Certain world events can lead to a rush in buying or selling of a given stock, in these situations it would be interesting to study the event in a reduced time scale. That is, if usually a given stock is monitored by the day then when the volume changes drastically the monitoring could change to minute-by-minute. This would be achieved by making use of time-scales. Time-scales represent the delay in which a moment is studied. For example if the time-scale is set to "hour" in a given moment, the next event will be retrieved from the database where the timestamp is increased by one hour, if the time-scale is set to "minute" the increase will be by the minute.

3.4.3 Parameterization

So that the tool can adapt to different scenarios and types of input, several parameters are available.

- Products - a string that contains the name of all the products to be compared.
- Value Types - the data value type should be set along with the relation value type so that the algorithm can get from the database the relation operator and the correct data.
- Time interval - a start and end time for the period to be analyzed.
- Delta (δ) - the maximum number of days between two events.
- Thresholds - the upper and lower limit for event recognition can be set independently.
- Influence Type - either Buy, Sell, Counter-Cycle or All (Buy + Sell). For ease of computing these are codified as 1, 2, 3, and 4 correspondingly.
- Edge type - sets the output tree so that it has its edges' weight as a percentage of total events or as the number of events.

These parameters along with the database structure make the tool very flexible and agile in terms of possible combinations of data, as long as the information is in the database the tool can compare it. Only two of these parameters need to be optimized for the best results, δ and the thresholds.

The parameter δ is usually only used if there is already an expected time restraint, in other cases it probably should be $+\infty$ so that it doesn't interfere. As for the thresholds unless a specific study is being done for a certain value, there is a need to find the thresholds that produce the best balance between a good percentage of positive events along with a high number of occurrences.

Due to the way that influences are calculated and how the Back-And-Forward heuristic builds the final tree, any small change in any of these parameters can greatly alter the

final result. This doesn't mean that the result of one parameterization is more correct than another. It just means that by changing parameters, the focus on the reality observed in the data is done slightly differently and that is enough to cause a cascade of minute differences that ultimately lead to significant changes on what is seen in the final output.

3.4.4 Influence Type

It's already been stated that there are four types of influence available - Buy, Sell, Counter-Cycle, and All - each one represents a different way of looking at how products influence each other.

The purpose of Buy is to find patterns where one product rises above the upper threshold after another did the same within the past δ days. This choice will be the most useful for the stock/bond market as it will be able to alert of products that often increase in value due to the influence of other products. Buy can then also be referred to as "increasing pattern".

Sell will have the opposite usage and behavior of Buy, it will look for events where one product falls below the lower threshold after another did the same within the past δ days. This behavior is useful as a defense mechanism in a way that it will detect products that act as early warning for big losses in the market. If a close watch is kept on the products that have a lot of heavy outgoing edges, losses can be minimized by dumping products that are usually influenced by the falling product. As the opposite of Buy, Sell will be then the equivalent of "decreasing pattern".

The third type, Counter-Cycle, will find events where one product rises above the upper limit while another stays below the lower threshold or vice-versa. This is useful to find outliers in the group, any product with a lot of edges in this influence type is probably not close to the rest of the products. However if the weight of an edge is consistently high, than the product might be inversely related to another. Depending on the behavior, Counter-Cycle can help the same way as Buy, Sell, or simply to identify outliers.

Finally, All combines both Buy and Sell so that the focus is put on products that influence others both on increasing and decreasing market status. This way trend setting products can be identified, that is, products that rise and fall before others do. It would be expected that major products in their relevant sectors would often be found to have heavy average edge weights when using this influence type.

3.4.5 Confidence Mode

The base version of the Ramex Forum algorithm keeps track of the number of events in an integer value, however this metric is not very easy to take conclusions from in terms of influences. For example, an influence weight of 3 days might be good in a period of one week, but if the analyzed time span is a month it probably would be considered bad. Furthermore, how do those 3 days fit in the number of expected influences?

When looking at *Buy* we expect that when one influencing product rises above the limit (upper threshold) that soon after its influences will also rise, but some times it will take longer than usual for that to occur or it never does. The same happens with the other influence types with their differences in event detection. It was then created a counter that will keep track of how many times each product is in a state that can result in influences, this counter will then be used with the number of detected events to calculate the percentage of actual events within the expected ones. This percentage has previously been referred to as *Confidence* in Section 3.3.1.

When this mode is selected the output will be the same as in the normal operation mode, but instead of seeing the edge weight as a number that represents the amount of detected events it will be a percentage that represents how many expected moments of influence were actually influences. This way if looking at *Buy* and one edge from A to B has a value of 70% the user can read this as "70% of the times that A rises above the upper threshold B will also do the same".

This is a very useful metric but it doesn't replace the event count because some times the number of events is also important, it's not very useful to know that A influences B 70% of the time if that only happens once a year. Because of this, both are available for use.

3.4.6 Algorithm Output

The same information can be presented in several different ways, in the case of the projected tool the data to display is a series of nodes connected to each other by weighted edges.

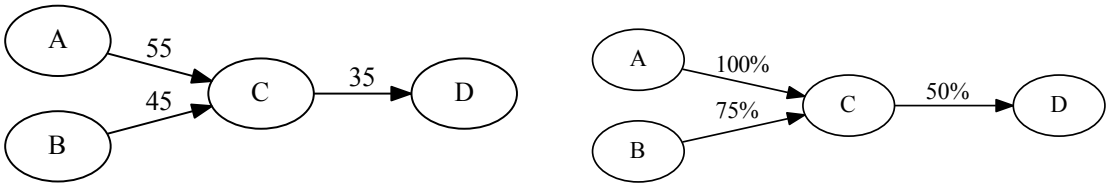
The algorithm creates an instance of the class Graph that represents the final tree, the output is then just a collection of objects in the computers' memory. This is not something easily readable by humans, and readability is one requirement for the tool.

The Graph class was then coded with several functions that write textual representations of the output in a easily read fashion. An example of that can be seen in Listing 3.1 where each line represent an edge between two nodes and the label represents its weight.

Listing 3.1: Textual output example

```
1 "A" -> "C" [label = 55 ];
2 "B" -> "C" [label = 45 ];
3 "C" -> "D" [label = 35 ];
```

However, for big quantities of information, reading that textual representation isn't optimal. With the aid of Graphviz an image is produced that displays the relations in an intuitive way. Figure 3.4(a) and Figure 3.4(a) show two examples of the final output with the two modes of Edge type seen in 3.4.3.



(a) Event Counter (b) Event Percentage

Figure 3.4: Examples of the graphical output of the algorithm for the two types of graph available.

This figure is a great example to show how one can read these graphs, on the left each edge's weight shows the number of observed influence events from one node to another. On the right, each edge's weight show the percentage of detected events within the expected events.

For example, the edge $A \Rightarrow C$ has an event count of 55 and a percentage value of 100, this means that there were 55 expected events and all of those turned out to be actual events. For the edge $B \Rightarrow C$ with an event count of 45 and a percentage value of 75, this means that there were 60 expected events and only 75% of those turned out to be actual events.

3.5 Space and Temporal Complexity

In this section the space and temporal complexity of the algorithm will be discussed, this study is needed so that it is known what to expect in terms of computation time and scalability.

3.5.1 Space Complexity

The algorithm primarily works over an instance of Graph, adding new edges as they are detected. However the detection of events and keeping track of their values require some secondary arrays.

Each Graph keeps information regarding all its edges and nodes in vectors of the classes Edge and Node correspondingly. Lets consider the number of edges as E and the number of nodes as N. These two vectors will then have a space complexity of $\mathrm{O}(E)$ and $\mathrm{O}(N)$ correspondingly. A complete graph has $N\frac{N-1}{2}$ edges, this means that at the worst case scenario the space complexity of the vector of edges is $\mathrm{O}(N\frac{N-1}{2})$.

To reduce temporal complexity in the search for edges, a matrix that keeps track of existing edges in the form of

$$[source\_node][destination\_node] = edge\ position\ in\ the\ edge\ vector$$

was added to Graph. As expected this matrix will have a size of $O(N^2)$, a Graph will then have a space complexity of $O(N\frac{N-1}{2} + N + N^2)$.

In order to build a Graph from the information in the database four arrays are needed, all of them with a size of N. These arrays contain the number of times each product is above/below the threshold, the number of consecutive times where each product is above/below the threshold and the value of each product for the current timestamp.

The total space complexity of the algorithm will then be $O(\frac{9N+3N^2}{2})$ which can be simplified to $O(N^2)$.

3.5.2 Temporal Complexity

Ramex Forum is split into two stages as discussed in Section 3.1, the first one will build the initial Graph from the information in the database and the second will turn the graph into a tree. The temporal complexity of the most used functions of the class Graph and for each stage of Ramex Forum can be seen in Table 3.1.

Most functions implemented in the Graph class have constant complexities, with the exception of initializing the class, outputting the graph and finding the heaviest edge. Improvements can be made to optimize the initialization and search for the heaviest edge, in the first case by not initializing the *edge exists* matrix with a value of -1 and using 0 as the indicator of non-existing edge and in the second case by keeping track of the heaviest edge any time an edge is inserted, modified or deleted.

Now for the Ramex Forum algorithm itself a preparation stage also needs to be considered. To simplify the way the data returned by the database is processed, it is important that the list os products is in alphabetical order. This isn't considered in the first stage because it could be defined as a pre-condition for the algorithm and ignored, however it is important to know that this needs to be done and what its cost is. Since Ramex Forum will work over a Graph, it first needs to be instantiated with a heavy cost, this burden could however be alleviated as discussed previously.

With everything ready, the algorithm can start processing. The first stage will read the values for each node in every time unit available and calculate their status, then for each time unit the nodes will be compared to each other to assert correlation between them. The first stage is where most of the workload will happen, both in terms of the algorithm and database accesses. The second stage starts by finding the heaviest edge in the graph, after that all edges will be iterated for every node to find the heaviest edge that doesn't generate cycles. The algorithm's complexity is then quadratical and has a complexity of $O(TU * N^2)$.

Table 3.1: Temporal complexity of the most used functions of the class Graph and each Ramex Forum stage. E = number of edges; N = number of nodes; TU = number of time units.

Function	Complexity
Initialize	N^2
Write To File	E
Add Node	1
Add Edge	1
Remove Edge	1
Decrease Edge Weight	1
Get Edge	1
Get Heaviest Edge	E
Edge Exists	1

(a) Graph

Function	Complexity
Preparation	$2N^2$
Sort Products	N^2
Initialize Graph	N^2
First Stage	$TU * (N + N^2)$
Process Moments	$TU * (N + N^2)$
Second Stage	$E + N * E$
Get starting heaviest edge	E
Find heaviest for each node	$N * E$

(b) Ramex Forum

The second stage will look at the Graph built in the first stage and will start searching for the heaviest edges of each node that don't cause cycles, thus generating a tree. The process is started by picking the heaviest edge in the graph, from then for each node all edges will be iterated to find the heaviest one that connects to the current node and doesn't connect to another node already in the output tree. This process has a time complexity of $O(E + N * E)$.

3.5.3 Scalability

Retrieving the values for all the nodes in each time unit, calculating their status, and comparing them to each other is the most time consuming action in the algorithm. This is caused by two problems, the first one is that accessing the disk to retrieve the values takes a lot of time. And when the analysis is done over a large period, the time spent waiting for data to be retrieved really adds up. Second, a quadratical complexity on the nodes means that a very large analysis might be become unfeasible fast.

The first problem can be partially alleviated by pre-calculating the status of each pair of data-relation value type and keeping it in the database, this will mean that less time is spent retrieving data from the database because only the status will be retrieved. This is especially useful in cases where the same product list is ran with different parameters, the first time the list is ran the status is calculated and inserted in the database and subsequent runs will greatly benefit from this, in some cases the time is reduced to almost a quarter.

As for the second one, it outside the scope of this thesis to find a way around it. The purpose of the algorithm is to compare the behavior between all the products in the input

and for that a comparison must be made between every pair for each studied moment.

3.6 Algorithm Validation

In order to validate both the algorithm itself and the implementation, a series of tests were planned and executed. These tests need to evaluate whether the parameters have the correct effect in the output and if the expected influences and relations show up in the output.

3.6.1 Simple Tests

A set of six very simple hand made tests were done to see if the four different influence types are being correctly detected. While the focus of theses tests is the influence types, δ and the limits can also be tested.

These tests consist of a small set of values for two products (A & B), each value representing a moment, that change in strict patterns where it would be expected that the algorithm will interpret as events. The first two tests will be for *Buy*, the third and fourth for *Sell*, the last two will be for *Counter Cycle*, and the first four will also be used for *All*. All the tests will be continual in time, one after the other and all the test values will be used for every parameter run, this way it can be seen that the algorithm detects the expected values and none of the others.

In the first test, B will rise above the upper limit one moment after A does the same and both stay there for one moment. It is expected that when in *Buy* or *All* mode this will create an influence of A->B with a weight of one for any value of δ equal or above one. The result can be seen in Figure 3.5(a) in green.

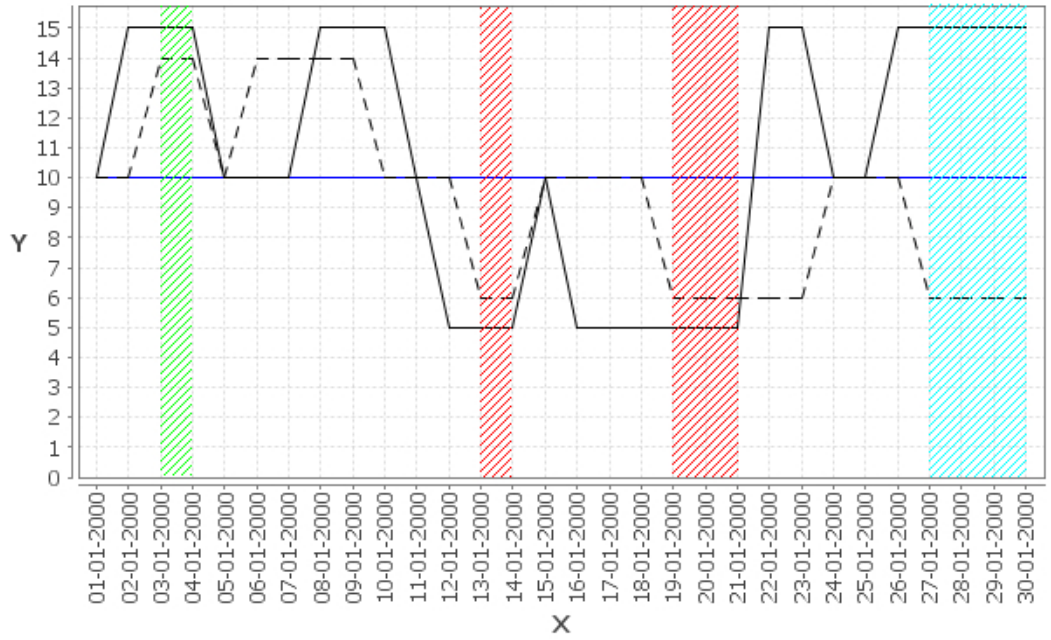
In the second test, A will rise above the upper limit two moments after B does the same, B will return to below the upper limit one moment later. It is expected that when in *Buy* or *All* mode this will create an influence of B->A with a weight of one for any value of δ equal or above two. The result can be seen in Figure 3.5(b) in green.

In the third test, B will fall below the lower limit one moment after A does the same and both stay there for one moment. It is expected that when in *Sell* or *All* mode this will create an influence of A->B with a weight of one for any value of δ equal or above one. The result can be seen in Figure 3.5(a) in red.

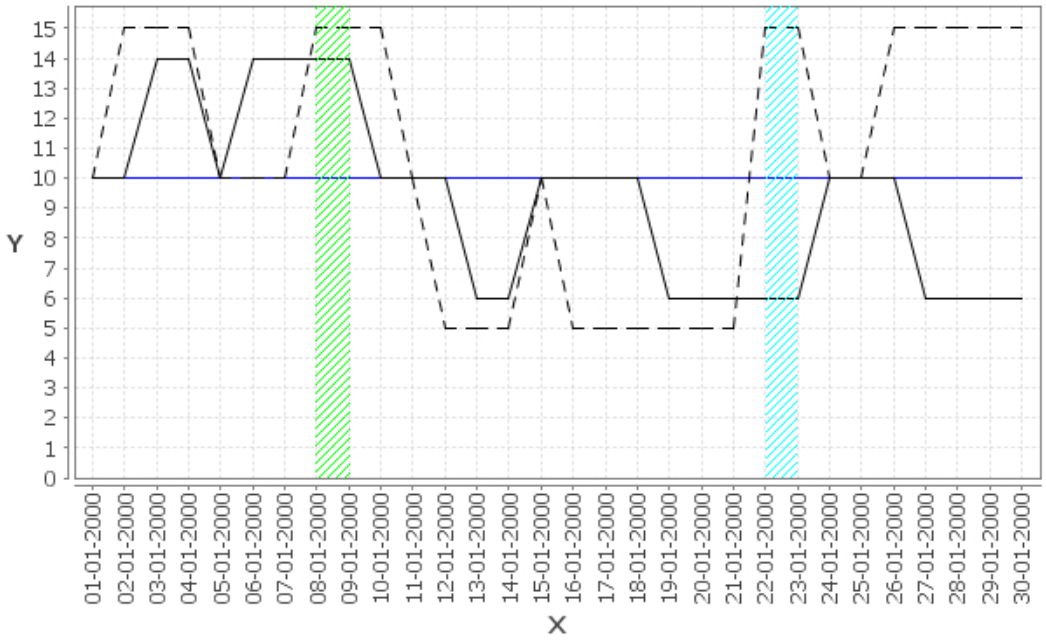
In the fourth test, B will fall below the lower limit three moments after A does the same and both stay there for two moments. It is expected that when in *Sell* or *All* mode this will create an influence of A->B with a weight of two for any value of δ equal or above three. The result can be seen in Figure 3.5(a) in red.

In the fifth test, A will rise above the upper limit three days after B fell below the lower limit in the fourth test and will stay there for one moment. It is expected that in *Counter Cycle* mode this will create an influence of B->A with a weight of one for any value of

δ equal or above three and no influences in the other modes. The result can be seen in Figure 3.5(b) in cyan.



(a) Influences from A (full) to B (dashed)



(b) Influences from B (full) to A (dashed)

Figure 3.5: Two graphs showing the test values and the algorithm result.

In the sixth test, A will rise above the upper limit and one day later B will fall below the lower limit staying there for three moments. It is expected that in *Counter Cycle* mode this will create and influence of A->B with a weight of three for any value of δ equal or

above one and no influences in the other modes. The result can be seen in Figure 3.5(a) in cyan.

The test values were loaded into the database and the chart result of eight runs of the algorithm with different parameterizations of *influence type* and δ were collected and can be seen in Figure 3.5.

As expected the results indicate that the algorithm behaves according to what was predicted, it can then be concluded that at a basic level the implementation is correct.

3.6.2 Generated Tests

The simple tests are good enough to test for expected behaviors, however datasets will not have such strict patterns and temporal developments. It was then decided that to further validate the implementation and the algorithm, a new set of randomly generated values would be used.

3.6.2.1 Test Set Generator

The first approach of using simple random values to generate two sets, taking their average and comparing them to each other wasn't very interesting. It was too unstable and there where rarely big enough sequences of continual above/below threshold values.

Because the algorithm finds sequential patterns, it's important that the two sets have a similar behavior with a somewhat constant time displacement. This could easily be done using two sine waves displaced in X but that would remove the unpredictability.

The next step is then to introduce some variations in both the amplitude and frequency of the sine waves, this can be achieved using the following function, adapted from [MG10]:

$$F(t) = A.sin(k * t) + B.sin(k * t)^2 + C.rand() \tag{3.1}$$

Two sine waves with different weights are combined and a random factor is added to introduce noise. The variable k is used to set the frequency and the function rand() generates values between -0.5 and 0.5 so that the noise can either increase or decrease the value of the function momentarily.

The test set generator was implemented in Octave because it allows to easily generate the sets, normalize them, calculate their moving averages, and to output the values both in graphical and textual forms.

Each set is output as a table of comma-separated-values (CSV) that can easily be imported into a database system. Along with the table, a graphic is drawn with the values so that the output can easily be evaluated by hand before being ran in the tool. This way it can be confirmed that what the tool shows is the same as what was generated.

3.6.2.2 Results

Two sets of values, A and B, were generated using the test set generator with the parameters seen in Listing 3.2 along with the number of days above/below the thresholds, how many days A and B are both in the same state at the same time, and how many days A and B are in distinct states at the same time(counter-cycle).

Listing 3.2: Octave output for the value generator.

```
Number of moments: 100; Moving average size: 40; Threshold: 5%; Delta: 10
# A #
Parameters -> A: 0.096473 B: 0.989559 k: 0.100000 C: 0.200000
Number of days above upper limit: 52 Number of days below lower limit: 45
# B #
Parameters -> A: 0.500954 B: 0.258241 k: 0.100000 C: 0.200000
Number of days above upper limit: 48 Number of days below lower limit: 52

Concurrent days above upper limit: 26
Concurrent days below lower limit: 24
Days in counter-cycle: 47
```

The values of the sets can be seen in the graphic of Figure 3.6, this graphic has 4 lines and 2 sets of symbols. The 4 lines are divided in two colors to distinguish between the sets A and B, in each pair one of the lines is the set of values and the other is the moving average. The moving averages can easily be identified by their smoothness and clear sinusoidal behavior, in the original graph they are dashed but the detail was lost when capturing the image.

Figure 3.6: Octave output displayed in a graphic. Black lines are A and A's moving average. Red lines are B and B's moving average.

The black crosses and red circles are used to identify the state of A and B respectively, in each moment. If the symbol has a value of one, it means that at that moment the set value is above the upper threshold(moving average + 5%), if the symbol is zero it means that the value is below the lower threshold(moving average - 5%), and if the symbol is at 0,5 it means that the value is between the upper and lower thresholds.

This way it becomes very easy to visually identify at what moments influences begin or end. Whenever both symbols overlap, by looking back along X until there is only one symbol in that line, the product that remains will be the influencer and the one that changes will be the influenced. This is a good way to visualize influences when the number of moments is small, but it loses effectiveness when long periods of data are analyzed, that is why it isn't used in the tool.

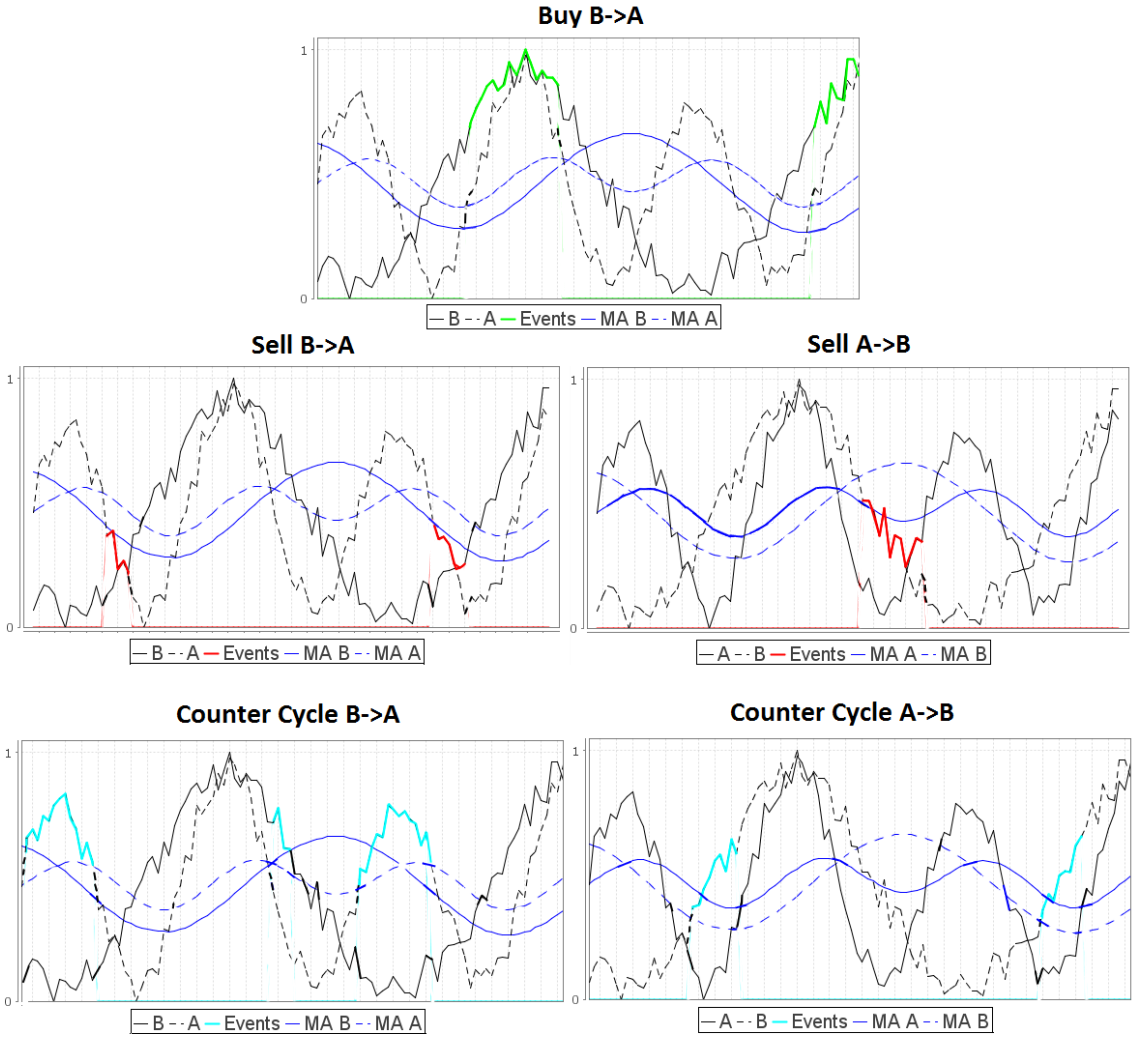


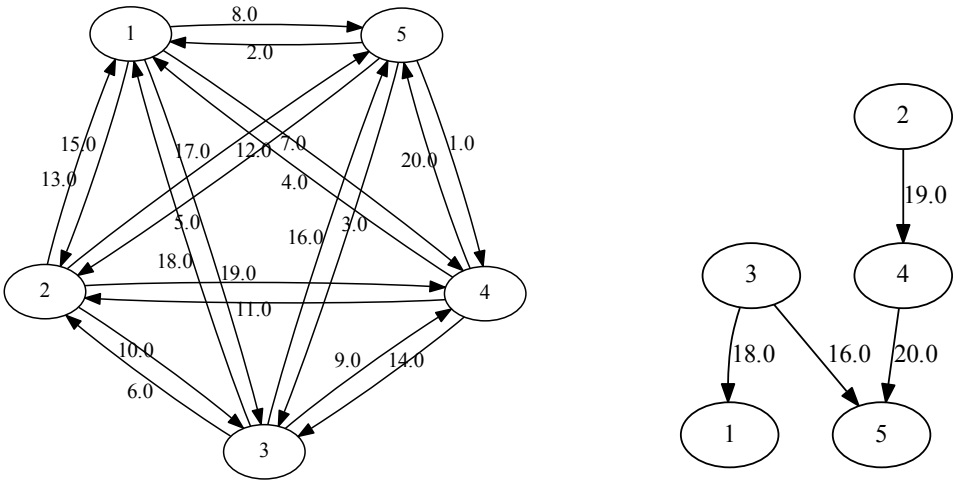
Figure 3.7: Results from the first stage of the Ramex Forum algorithm as seen using the tool for Buy, Sell and Counter Cycle.

The output of the tool for the generated test can be seen in Figure 3.7, periods where the algorithm detected influences are highlighted with green for buy, red for sell and cyan for counter-cycle.

It is clear that all events identified when the data was generated are also detected by the tool. The relevant sections are easily identified using the tool's graphic display and, even if not shown here, the event count also matches with what was expected. These outputs show that the implementation, algorithm, and the choice of display method continue to hold up correctly and has expected.

3.6.3 Validating the tree creation

The two previous tests only address the first stage of the Ramex Forum algorithm, for the second stage a complete graph with 5 nodes will be used to evaluate how the Back-and-Forward heuristic behaves. The edges between nodes will be randomly assigned a unique weight between 1 and 20 (a complete digraph with 5 nodes will have 20 edges), the purpose of using unique values is so that conflicts between equal weights are avoided.



(a) Input (b) Output

Figure 3.8: The complete digraph (a) that was used to test the second stage of the Ramex Forum algorithm and the resulting tree (b).

Given that the purpose of the second stage is to take a single connected tree from the input graph, using as input the graph seen in Figure 3.8(a) the steps taken to reach the optimal tree are: 1 - find the heaviest edge and add it to the output; 2 - keep adding the heaviest edges that don't cause cycles until all the nodes are connected in a single output graph as seen in Section 6.

With those steps done, the result will be the same as the output seen in Figure 3.8(b), showing that the second stage of the algorithm is working correctly. While not the most extensive testing, this example is enough to see that within the expected working environment the algorithm behaves correctly.

3.6.4 Observations of the algorithm validation

With the previously discussed tests, it was possible to test that the basic functionalities of the algorithm work as expected, both with theoretical and practical examples. The patterns between pairs of products were correctly found and the parameters had the expected outcome when changed.

Because the analysis is done in pairs of products, increasing their number will only incur in repeating the same process for each pair. It can then be safely expected that if the algorithm behaves well for one pair, it will work well for all subsequent pairs. Because of this no validation was done with larger groups of generated tests, the basic algorithm would work and any influence pattern found between the generated products would be meaningless and a waste of time to evaluate.

3.7 Simulator

Given that the output of the algorithm tells us that in the past some products influence others, or at least appear to, these observations could be used to try to predict future behaviors. With that in mind it was important to test if this was true or not. For that a small simulator was built that, given historical price data, calculates the influences between products and buys or sells them based on the influences found, keeping track of how well each buy or sell choice turned out.

For each day the simulator will use the first stage of Ramex Forum to calculate the percentage of influence events between all products. Then it will check which products are above/below the upper/lower threshold and buy/sell any products that are influenced by them. Thus keeping a portfolio that contains only products that are influenced by other products, with these being recently above the threshold while none being below the lower threshold. Each time there is a buy signal the simulator will see if there is still a free parcel of cash that can be used to buy the product, if there is it buys the maximum amount possible with the cash value of a parcel. If there are more buy signals than available parcels the products are bought in a first come first serve basis.

A simple control heuristic was also included that buys a product whenever it is above the average and sells it when it falls below the average. The results obtained from the Ramex Forum predictions will be compared to this so that there is some performance comparison. If the signals from the algorithm perform worse than this simple heuristic then it probably isn't very useful.

For each simulated day a number of metrics is taken so that the performance evaluation can be done. These metrics are the number of bought products, the number of bought products that were a good buy, the control bought products, the control bought products that were a good buy, and the number of products that is above their average. With them the percentage of correct buy signals for both the algorithm and the control can be calculated and compared, the final metric gives an idea of the current market trend (how many products are in a rising trend). Taking these values and plotting them along with their average in a graph, for each day, shows how well both signals work in terms of correct/incorrect buys in a clear way. It is important to note that the market is defined as all the products that are currently being simulated and not the whole financial market.

A buy order will be considered a good one when in the next δ days there is at least one moment where that product's price rises by X% above the moving average. X will be the upper threshold defined for the Ramex Forum algorithm. This doesn't always translate to a price increase, the price can fall fast enough to bring down the moving average, and then come back up to a price lower than the original but still X% above the moving average.

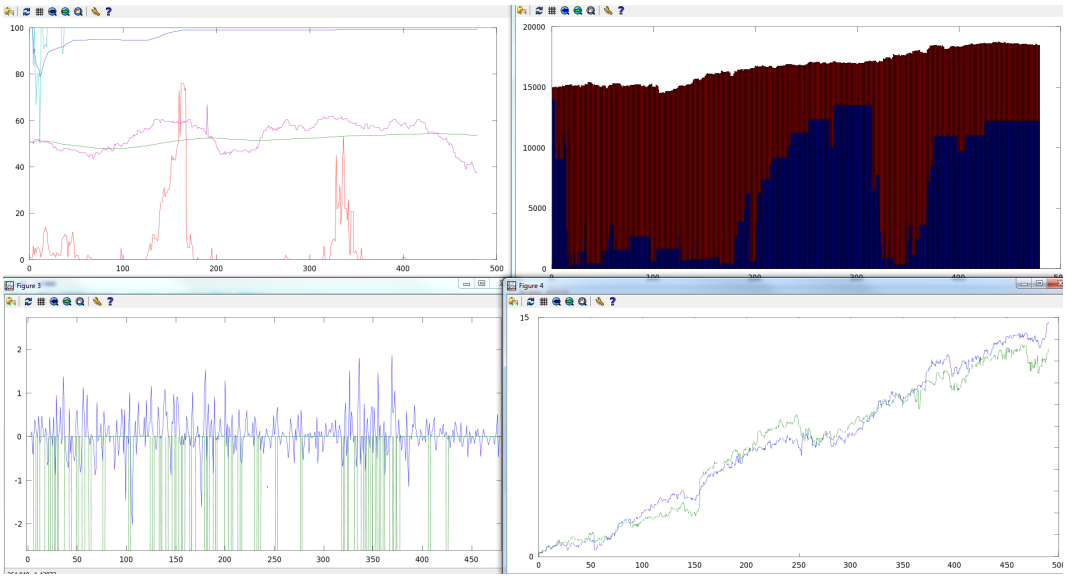


Figure 3.9: Simulator output example.

Only correct buys are looked at because the main objective would be to gain money instead of protecting against falling products, if the simulator proves that the algorithm works well for buy orders then in the future an evaluation of sell orders can also be done.

However in the real world just a percentage of correct guesses isn't enough, actual values of gains or losses need to be evaluated, the composition of the portfolio and available credit needs to be taken into account. The first step was to define the starting capital and into how many parcels it should be split between products, the first value was set at a conservative amount of €10.000, the second is harder to define.

If the number of parcels is too high, individual price changes will have a small weight

on the price, and in times with few buy signals a large part of the money will just sit there. If the number of parcels is too low, one product bought at the wrong time will mean massive losses. On top of that, the middle ground between these two options is not clear, some periods have a lot of buy signals while others have long bouts of no signals. It was then decided that for periods with a smaller number of buy signals the parcel count will have a value of 10 and in the other case it will have a value of 20.

In order to evaluate the progression of market behavior, wallet value, and correct orders, four different graphs are drawn as the days are simulated, an example of these graphs can be seen in Figure 3.9. The first one shows the previously mentioned metrics and has four lines: 1) the percentage of correct buy orders for the day; 2) the average of correct buy orders for the whole observed period; 3) the percentage of correct control buy orders for the day; 4) the average of correct control buy orders for the whole observed period; 5) the number of market products above their moving average for the day.

To keep track of how much of the wallet's value is cash or products, a stacked bar graph is used. For each day, the sum between cash and the value in products will results in the wallet value, the cash portion is shown as green fill and the product portion is the orange fill. With this simple bar graph two things can be observed, first, how the wallet value progresses, and second, a broad visualization of how the algorithm is reacting to the market with buy and sell orders.

The simulator gives sell orders based on three situations, either because the algorithm signals that a product's value is probably going to fall soon, or because a product valued a lot (limit set at 5%) and it's best to just sell it right away so that its value doesn't fall. Or due to the use of a stop-loss setup where if a product's value falls too low (limit set to a more conservative 2,5%), it is immediately sold to prevent further losses. This lead to the question of exactly how successfully these orders are behaving, for that a third graph is used. This graph shows, for each day, the result of all sell orders in percentage of gains/losses for both the algorithm and for the control. There isn't much information that can be gathered from this graph, it's mostly used to compare behaviors of different parameterizations as sort of a debug feature.

Finally, the fourth graph shows the percentual change in value in relation to the first day of simulation of the wallet, the control wallet, and the total market value. This way it's easy to evaluate how the algorithm behaves in relation to the control and the market. This is the fastest way to evaluate the utility of the algorithm, it must at least be able to get a higher value than the control and also fare better than the market total. As long as the Ramex Forum line stays above the other two lines, it indicates that the algorithm is providing adequate buy and sell signals.

The data used for the third case study (Section 5.3) was the best candidate for simulation since it has a large number of products, from varied sectors, that showed to have strong connections between themselves. The results obtained with the simulator will be further discussed along with the results for the case study.

CHAPTER 4

THE GRAPHICAL TOOL

Console based programs aren't very user friendly and given that the target audience for this kind of tool would be the usually less tech savvy economists, a simple graphical user interface (GUI) was developed.

The purpose of this component isn't to create a package that would be ready for mass deployment with a very complete and thought out interface, but simply to ease the parameterization process and facilitate access to the information generated by the Ramex Forum algorithm.

Every single display functionality that will be discussed in this chapter can be replaced with already existing software such as image and spreadsheet viewers. The basic tool already outputs all data in compatible text and picture forms, this GUI will just aggregate the needed functionalities into a single package and provide some extras that were found useful.

There are four different visualizations on this tool, this first and main one will provide fields and selectors to parameterize the algorithm and choose the input while at the same time displaying the output tree. The other windows are purely for data display with some inputs to control what information is displayed.

The Chart window shows a graphic with the history of prices for the selected pair of products along with the detected events between the same two products. This is useful to see exactly how those two products interact between themselves and in what days the influences were detected.

Because the data collected in the first phase of the Ramex Algorithm is impractical to display in graph form, the third window sums all the inbound and outbound edges of each product and displays that information in a table. This way the weight of each product can be evaluated without the losses of information brought by displaying the relations in a tree.

The final view, Focus, is useful in combination with the counters window as it allows

the user to see either all inbound or outbound influences for each product. This is useful if the user wants to know more about a specific product after checking the counters. For example if in the counter window one product is distinctly heavier than the others, the user can focus on that specific product and know what influences are causing that disparity.

4.1 Parameterization and the Graph display

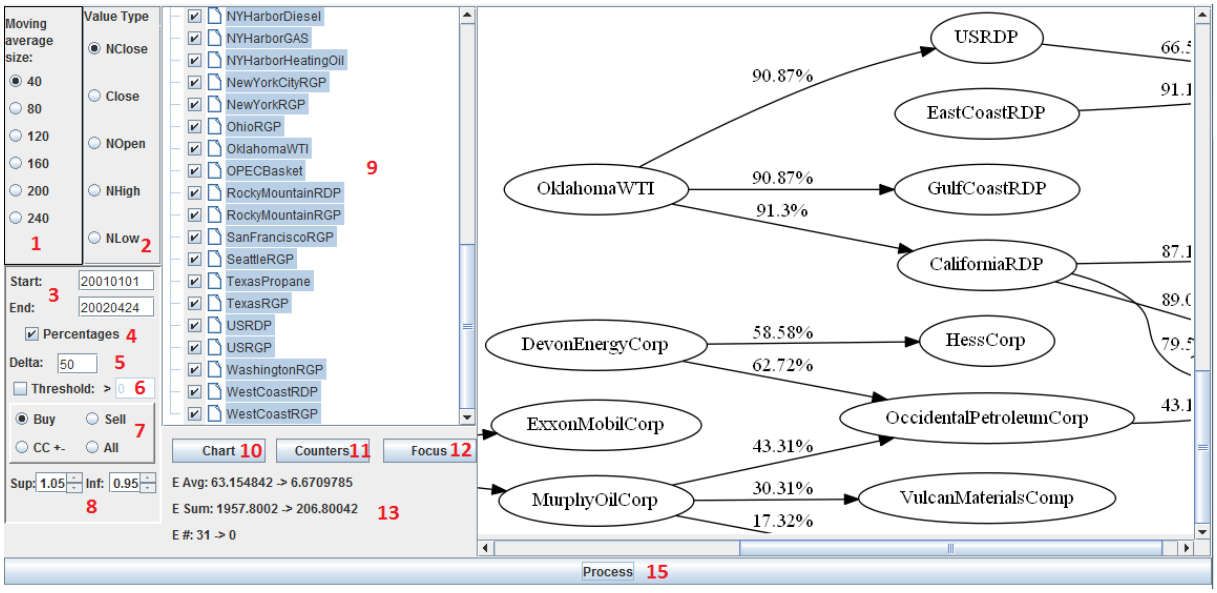


Figure 4.1: The tool's main window.

As described before, this is where the basic functionality of the tool is done and it will be here that the user will parameterize the algorithm and run it. Figure 4.1 shows how the tool's main window is organized and what its components are. It has fifteen components and in the following list the general function of each one is described:

- 1 - Moving Average Size Selector - allows the selection of one of the available preset sizes.
- 2 - Value Type Selector - allows the selection of one of the value types existent in the database.
- 3 - Time Interval - sets the starting and end date of the time intervale to be processed.
- 4 - Percentage Selector - how the edge weights are displayed, as a simple event count or as a percentage of true influences in the total expected influences.
- 5 - Delta - sets the value of δ.

- 6 - Weight Threshold - allows a minimum weight threshold where only edges heavier than that threshold are displayed.

- 7 - Influence Type selector - allows the selection of one of the four available influence types.

- 8 - Upper and Lower limits - sets the value of the upper and lower limits

- 9 - Product Selector - shows all the available products in the database and allows the selection of the ones that are to be analyzed.

- 10 - Chart Window - displays the Chart Window if a graph as already been processed.

- 11 - Counters Window - displays the Counters Window if a graph as already been processed.

- 12 - Focus Window - displays the Focus Window if a graph as already been processed.

- 13 - Resumed Information - shows the average edge weight, the sum of all edge weights and the number of edges. These values are compared to the last generated graph so that the difference between them can be evaluated.

- 14 - Result Display - shows the result.

- 15 - Process Button - starts the process with the parameters set using the previously described components.

4.2 Chart Window (Edge Focus)

In the chart window the user can select the pair of products to be displayed from two lists, the first list (1) shows only the influences visible in the final output of the algorithm, while the second list (2) shows all the influences found in the first stage of the Ramex Forum algorithm.

By selecting an influence, the related chart is immediately displayed in (9). From there the user can: change the timescale of the X axis using (3); enable the second operator (usually the moving average) with (4); enable the limit thresholds (5); enable an alternative method of displaying events (6); or change the start and end date of the charted data (7).

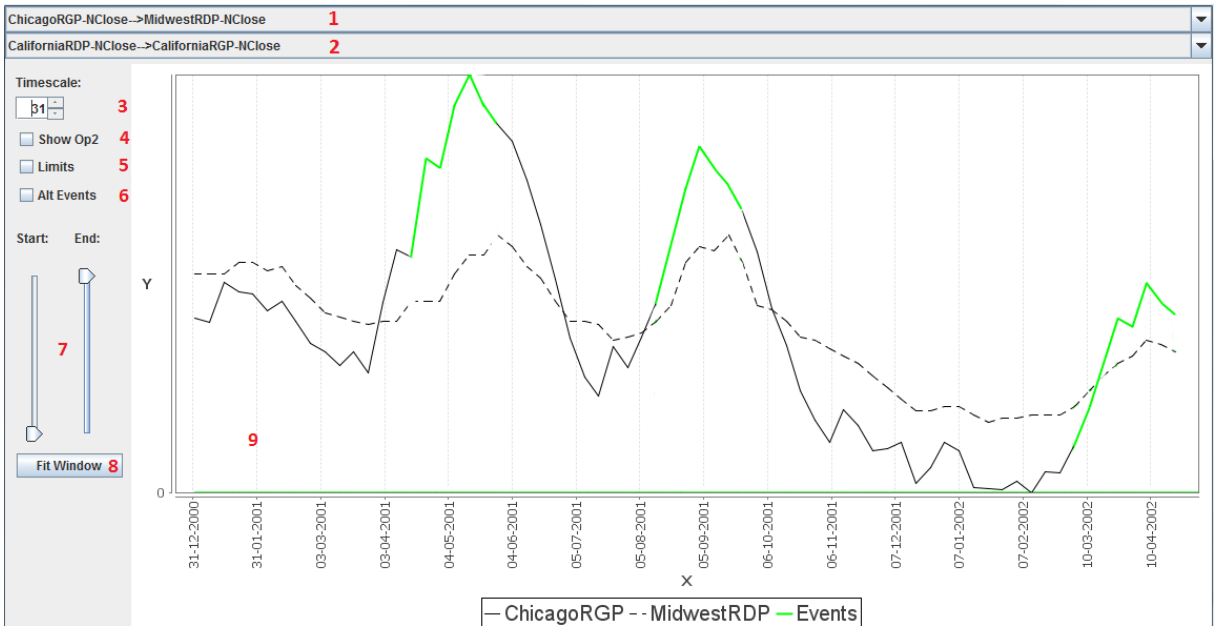


Figure 4.2: The tool's graphic display window.

The Fit Window button (8) will readjust the Y axis so that the graph lines occupy the whole chart area. This button is useful when paired with the ability to zoom provided by the chart component (9), the user can zoom in a certain area and then fit the selected area so that the chart space is fully used.

4.3 Counters Window

Here a simple sortable table is all that is needed to display the information, for each product the table shows the inbound/outbound sum or sum average (selected with 1) and the number of edges that make up that value. The rows of the table can be sorted by clicking the headers(3), making it easier to find the most influential or influenced products.

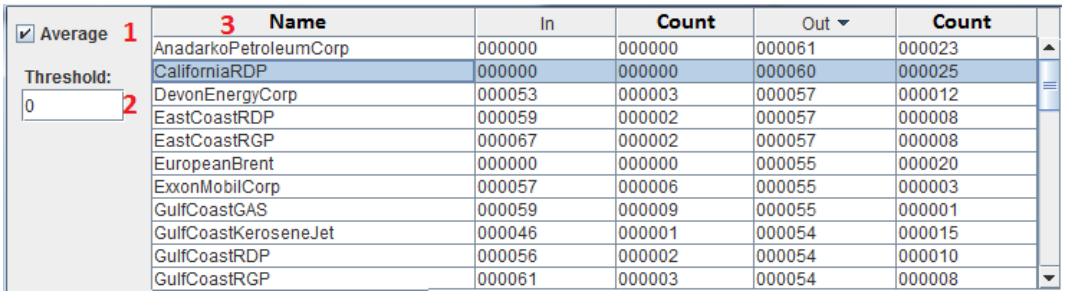


Figure 4.3: The tool's counter window.

Because values that are too low might be considered irrelevant and will influence the final results of the sum and average, a field to set a minimum threshold is available (2).

When this value is set only the edges that are above the value are considered.

As stated before, the values used for this table are those of the graph generated in the first stage of the Ramex Forum algorithm, this means that what is seen in this table might not reflect directly on what is seen in the final result of the algorithm. However this doesn't mean that the information gathered from both sources invalidate each other.

4.4 Node Focus Window

The focus window is basically the same as the counters one but instead of showing the information as a sum of all inbound/outbound edges, this one focuses on a single product and shows all the edges connected to it.

This is especially useful because it clearly shows a lot of information that is discarded when looking only at the tree and partially omitted in the counters. Here the user can clearly see who influences or is influenced by a certain product.

The user can pick what product to focus on with the drop-down (1) and the rest functions the same as the Counters Window with an extra button (2) that will show what is seen in the table but in graph form.

In the new drop-down (3) it's possible to choose the product and whether to see the inbound or outbound edges.

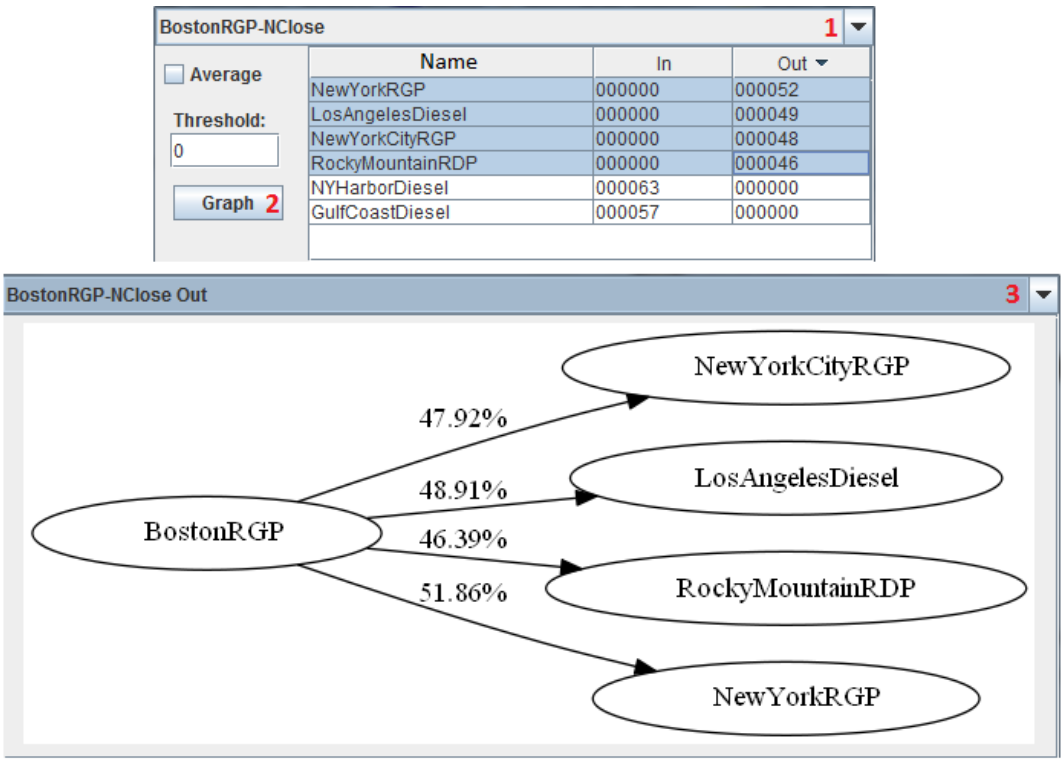


Figure 4.4: The tool's focus window.

CHAPTER 5

CASE STUDIES

One way to evaluate the utility of the tool is to perform case studies. A case study tries to analyze, evaluate and explain what happens in a given event or situation. Data is collected about several parameters in the selected case using a variety of sources and procedures so that a complete picture of the situation can be made.

First a thorough definition of the case study is done: what is being evaluated, in what ways, what is already known, what are the expected behaviors. Then the study is performed according to the specified parameters and limits. Finally, the results are evaluated for their validity and reliability so that proper conclusions can be made.

In this instance, the tool will be used in different past real world cases and the results obtained will be compared to what actually happened following the studied events or how specialized market reports describe the cases.

Three different studies will be performed: one to evaluate the relationship between the price of petroleum and the price of its derivatives; one to see how the different currency prices influence each other; and a final one to determine if the tool behaves positively in cases where it is known that there is a strong relationship between elements, such as in investment funds, where the value of the fund is directly proportional to its components.

5.1 Case Study 1: Petroleum and its Derivatives

As one of the most important resources to the developed world, petroleum has a big influence on the markets. It can be processed and refined into several other goods and substances such as gasoline, kerosene, asphalt, paraffin, plastics and lubricants. Because of this, a case study on petroleum was chosen so that, using the tool, an evaluation can be made on how the price of derivatives are influenced by the price of the source material.

The price of petroleum and its derivatives isn't influenced simply by supply and demand; taxes, speculation, wars, costs in refinement and transportation all contribute in setting prices. With that in mind, it is already clear that there won't be a 1:1 price relation between a source material and its derivatives. Another relevant aspect is that because there is a lengthy refinement process, a significant increase in the price of the source material should only reflect in the price of its derivatives after a period equal to the time it takes to refine(usually within 3-4 weeks [Bor+96]).

5.1.1 Hypothesis

When a product is produced from a source material, its final price will be dependent on several things like production, distribution and marketing costs, taxes, and demand. These costs vary from product to product, however the price of the source material will always reflect on the final price [Suv+09].

It's then safely assumed that the price of the raw material will directly influence the price of the manufactured product, this is a case that fits well within what can be found using the tool. In order to have the best results in finding these influences, it helps if the final product isn't a combination of raw materials otherwise a new layer of complexity in influences is created.

As such, petroleum is a very interesting candidate because it can be directly processed into several products without much need for other materials. An added advantage of petroleum and its derivatives is that because of their commercial importance, their price and availability is well documented and accessible, facilitating the construction of the case study.

Petroleum is refined into the following major product categories [GH01]: Asphalt, Fuel Gas, Gasolines, Jet Fuels, Kerosenes, Oils, Greases, Waxes, Cokes, Lubricants, and other Chemicals. This is a relatively extensive list where each category might have hundreds of sub-products, that means that it would be almost impossible to reliably study the relations between all of them. The main problem would be that there is no publicly available repository of information for all of these products. It is then necessary to select a more restricted group.

It is expected that the price of crude oil will have a heavy weight on all other prices, from the price at the refineries to the retail price of all its products. Also the price at the refinery should affect the retail price on the surrounding area while fluctuations in city price will be faster than state averages.

5.1.2 Data and Information Gathered

A set of historical data is needed to proceed with this case study. The U.S. Energy Information Administration [Eia] is a good source for the needed data as it provided a very large repository of historical values for a wide range of petroleum related subjects. These subjects include prices, crude reserves and production, refining and processing, imports and exports, stocks, and consumption rates.

To simplify this case study, only the prices of the goods will be taken into account. However in an in-depth study of the relations between all the available data other interesting patterns would surely be found.

Because the source of information is specific to the U.S.A. the most complete sets of data were related to that same region. This case study will then focus on the price of crude oil, diesel, gasoline, propane, and kerosene in the USA as the most complete available datasets are about them.

As an added point, it would be interesting to see how the variations in the prices of these products affects the stock value of the companies that produce them. For this, eleven corporations dedicated to extracting, processing, and selling of crude and crude related products were selected and their stock value was compared to the previously mentioned products.

The data is separated into four categories:

- known benchmarks [Ham+08] for crude oil(West Texas Intermediate as OklahomaWTI, European Brent, and the OPEC Basket - an average of the oil price in 12 oil-exporting developing nations);
- refinery price for Gasoline(New York Harbor and Gulf Coast), RBOB Gasoline(Los Angeles), Diesel(New York Harbor, Gulf Coast, and Lost Angeles), KeroseneJet(Gulf Coast), Propane(Gulf Coast), and Heating Oil(New York Harbor);
- national(Retail Gas Price as RGP, Retail Diesel Price as RDP), state(StateRGP, StateRDP), and city(CityRGP, CityRDP) averages for regular gasoline and diesel;
- Corporation stock values

While not the most complete list, it contains several representatives of both products and production steps. There are three values for crude oil price, then for each product there is the spot price at major refineries and the retail price grouped by regions. This way it will be possible to study how the price of crude oil affects prices at national, regional and local levels while also studying the different production steps in distinct regions of continental USA.

The prices are separated into retail/bulk price and spot price, this means that for some items the price is taken from retail sellers and for other items it's the security price at that day. They are divided like this:

Spot	Retail Gas Price	Retail Diesel Price
EuropeanBrent	BostonRGP, CaliforniaRGP	CaliforniaRDP
GulfCoastDiesel	ChicagoRGP, ClevelandRGP	EastCoastRDP
GulfCoastGAS	ColoradoRGP, DenverRGP	GulfCoastRDP
GulfCoastKeroseneJet	EastCoastRGP, FloridaRGP	MidwestRDP
LosAngelesDiesel	GulfCoastRGP, HoustonRGP	RockyMountainRDP
LosAngelesRBOB	LosAngelesRGP, MassachusettsRGP	USRDP
NYHarborDiesel	MiamiRGP, MidwestRGP	WestCoastRDP
NYHarborGAS	MinnesotaRGP, NewYorkCityRGP	
NYHarborHeatingOil	NewYorkRGP, OhioRGP	
OklahomaWTI	RockyMountainRGP, SanFranciscoRGP	
OPECBasket	SeattleRGP, TexasRGP	
TexasPropane	USRGP, WashingtonRGP	
	WestCoastRGP	

The starting time-frame of available data ranges from the 1980's to 2006, this is because different prices began being tracked at different times.

5.1.3 Setup for the Case Study

The data for 55 prices was normalized using Min/Max normalization, moving averages with different sizes were calculated, and it was all inserted into the database for processing. The time interval chosen for analysis was from 2006 to August 2013 in order to make sure that every price will be taken into account for the whole analysis period.

In all case studies the data is normalized so that all values are between [0;1], this is especially useful when there is a need to compare the value progression of products. If the information isn't normalized, drawing a graph between a product that is valued in 10's of dollars against another one that is valued in the 100's make the comparison very hard. If they are both normalized, increases and decreases in valued are always proportional to each other, facilitating both calculations and the visualization of the data.

Because different parameter sets yield significantly distinct results, and because it would be impossible to compare and evaluate them all, it is first necessary to choose a parameter set and focus on its result. For this case study the price list, the start and end date, δ, comparison choice, and value type will be fixed while the best values for thresholds and moving average size will need to be evaluated. In this case study a focus will be put on the *Buy* comparison because the price of oil and its derivatives rarely fall and the increase/decrease of prices is very asymmetrical with a strong lean towards increases.

The parameter δ was fixed at a value of 30 days, which translates to around 6 weeks because the prices are only taken on work days. Two weeks more than the expected as

seen in the hypothesis so that even if that value is wrong, enough events can still be caught without overextending the influence range.

Even with the value set, there was still a need to verify whether the hypothesis was true or not, so an analysis on how the average edge weight changes with increasing values of δ was done and the result can be seen in Figure 5.1.

The chart shows the average edge weight change in percentage for each increment in the value of δ. The X axis is the value of δ and the Y axis represents the percentual change from the previous value of X. Each line represents the results obtained using a different moving average size of 40, 80, 120, 160, 200, or 240 days.

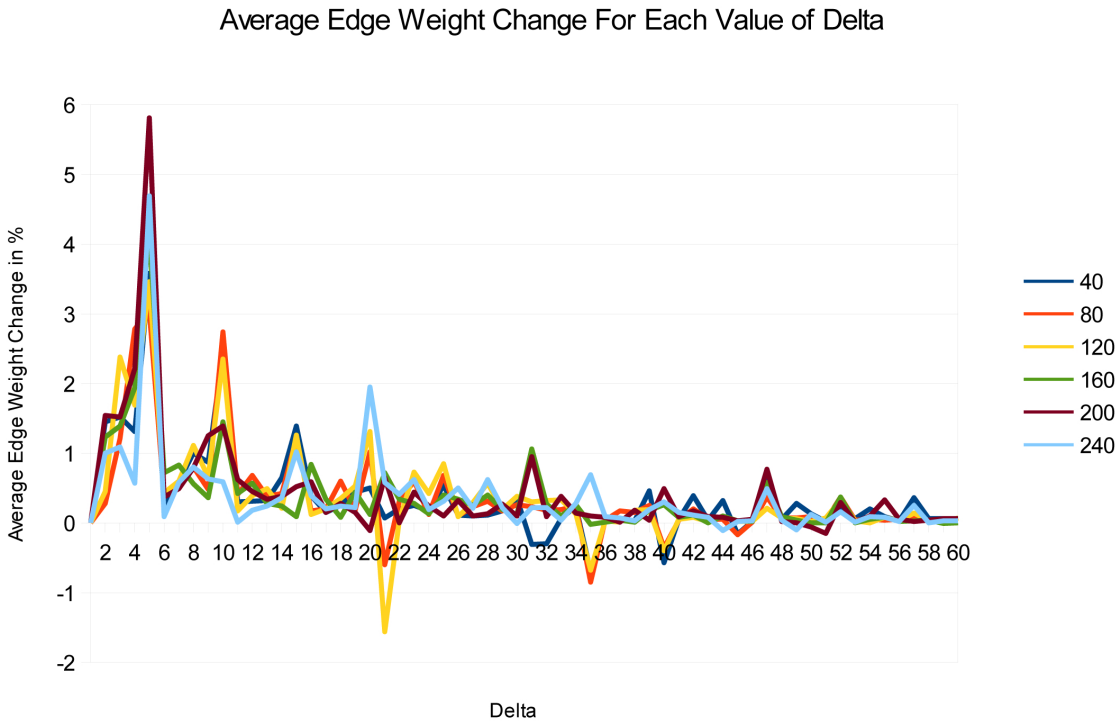


Figure 5.1: Graph showing the change in average edge weight with each increment of δ using the *Buy* comparison.

Several big spikes can be seen every 5 days, this is because gas and diesel prices at the pump are only registered on a weekly basis, so for each 5 day increase in δ the algorithm will pick up another change in value. This makes the analysis somewhat harder but it's still useful as now changes in retail prices are clearly identified by the spikes seen every 5 days.

The first thing noted is that at the first week there is already a noticeable increase in the average edge weight, however some of it is due to influences between retail prices and not only from refinery to retail prices. Second, after the fourth week the individual

increases in δ barely produce a meaningful increase in value, still the cumulative increases are significant. With these two observations and the information taken from the graph, the hypothesis that refinery prices affect pump prices in 3 to 4 weeks should be changed to something closer to "Refinery price changes usually significantly influence pump prices up until to four weeks.". As in, usually the influence can be immediate or it can take up to 4 weeks.

In order to find the best combination of parameters the algorithm was ran several times for a smaller time period (2010-2013) so as to not take an unfeasible time to calculate, and the edge weight average was recorded for each run. Taking the edge weight average consists of summing the weight of all the edges in the output of the first stage of the Ramex Forum algorithm and dividing that sum by the number of edges in the graph.

A spreadsheet was then made to combine and analyze the results. The best values for threshold interval and the moving average size were found using the graphs in Figure 5.2. All three graphs seen in the figure are important in choosing the best values for the parameters.

The first graph shows the progression of the average edge weight, in confidence mode (Section 3.3.1), in relation to the increase in threshold size. The parameters that lead to the highest increase in average weight can be clearly identified as the moving average size of 240 days with a threshold of around 26% of the moving average. At first glance this is a very good setting, the average weight in confidence mode increases by almost 110% in relation to using only the moving average as the threshold.

Things change when the event count is also considered, increasing the threshold rapidly decreases the number of detected events, a threshold of 26% will reduce the number of events by about 80%. In this case the starting average is around 130 events and falls to 30, in the 3 years period analyzed this translates to a very low average number of events. A choice needs to be made on what is prioritary, maximize either the certainty or the volume of the detected influences. The first choice allows conclusions like “We are very confident(>90%) that A influences B when the conditions are met but this happens rarely.” while the second will lead to conclusions more in line with “There are a lot of moments where A influences B but the consistency of this happening in the same conditions is low.”

For this case study the choice was made to maximize the volume so that a broader spectrum of influences can be detected instead of restricting the analysis to situations where the prices rise or fall sharply(which is what higher threshold values restrict the analysis to). Even with that, going for a threshold of 0% is not the best choice. For a moving average size of 240 days, increasing the threshold by 1 % will raise the average weight in confidence mode by about 5% while only lowering the event count by close to 2%, this trade-off favors the usage of this value.

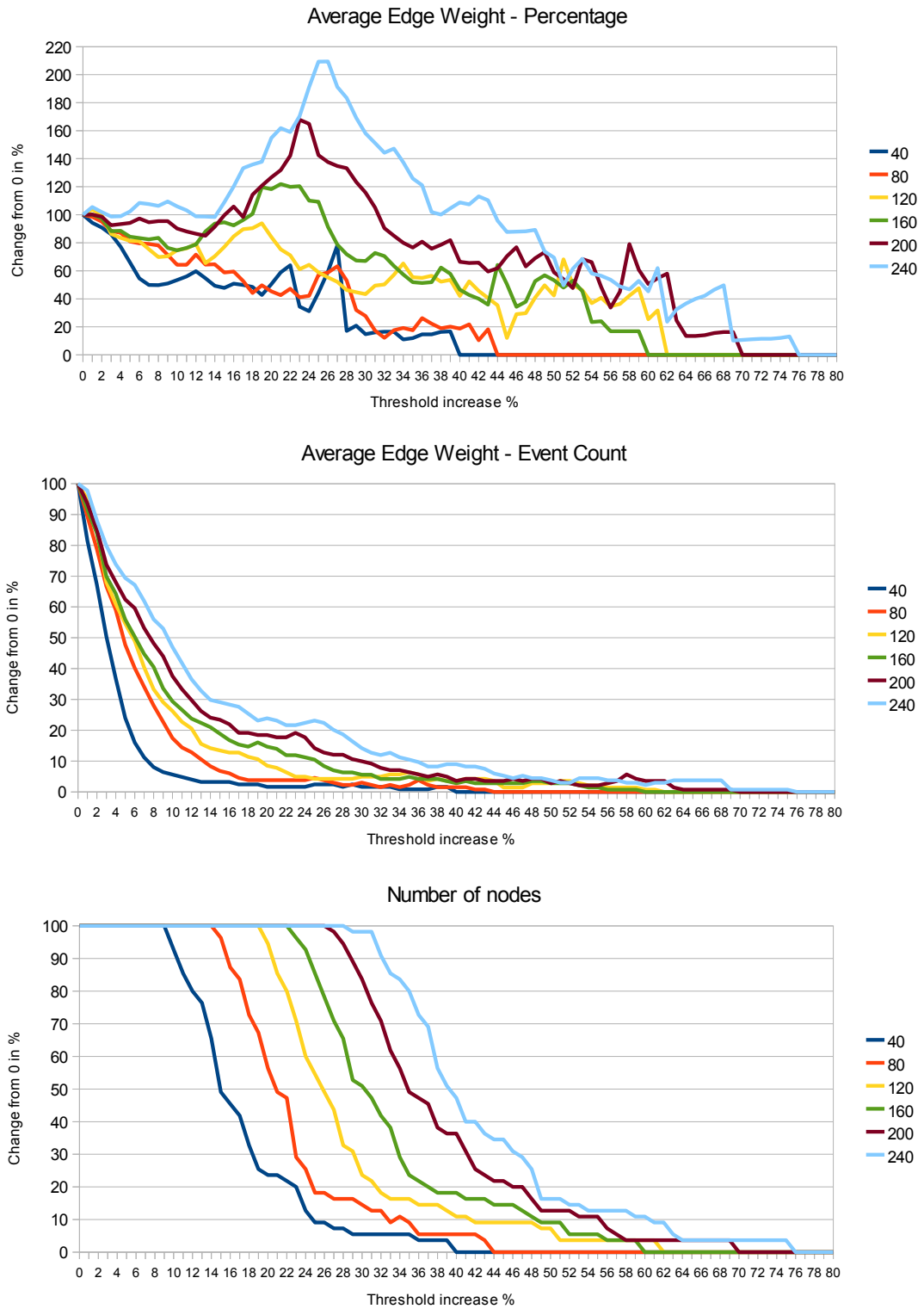


Figure 5.2: Graphs showing the change in average edge weight and number of nodes with each 1% increment in the threshold interval for $\delta = 30$ using the *Buy* comparison.

With the threshold defined, it's time to look at the moving average size. The choices for available sizes were based on [CM13] and from the graphs it's clear that maximizing this parameter yields the best results and even raises the question of how further increases in the size would fare. The user still needs to take into account of what it means to increase the moving average size, the bigger it is the smoother the curve will be and thus it will behave like a noise filter, becoming less and less sensitive to small changes in the behavior of the product.

Because the lines in the graph overlap, it's hard to read that in the first increment the 240 and 120 sizes have a very similar behavior. And because the graphs show the change percentage from the first value, it doesn't show that the average weight for a moving average of 120 days has a higher starting value than the 240 days one, this means that with a threshold of 1% the best choice for moving average size is 120 days.

For the *Buy* influence type the parameters were set at a threshold of 1% and 120 days for moving average size. The same study was done for the *Sell* influence type and it was found that the shortest moving average size of 40 days with a threshold of 0% yielded the best results. The shorter moving average size is probably justified by the market's quicker reaction to falling prices and thus a need for a more reactive limit.

Table 5.1: Defined parameterizations.

Influence Type	Threshold	Moving Average Size
Buy	1%	120
Sell	0%	40

5.1.4 Results

Using the parameters previously defined, the tool was used to produce the output seen in Figure 5.3. As previously stated, the tool uses Graphviz to create the visualization of the graphs and as such their spacial organization is done automatically. The only manipulation done over the image was to color the nodes to distinguish the different types of products.

There are some interesting pieces of information that can be attained, by closely studying the graph, that attest to its possible utility in finding more than just sequential patterns.

First of all, the colors show a clear grouping of product types, nodes with the same color are mostly close to each other. This was expected for gas to gas and diesel to diesel influences but even the stock, refinery, and reference benchmark prices tend to group together at least in pairs. Furthermore, refineries are almost exclusively related to the same type of product, gas producing refineries are connected to retail gas prices and diesel producing refineries are connected to diesel retail prices.

The GulfCoastGAS node doesn't exactly meet the previous observation as it is shown influencing some diesel products, even so this might be a positive thing as it will alert an attentive user to the weight behind the Gulf Coast refinery gas prices. Using the Counters function of the tool on this case study, the GulfCoastGAS is identified as the most influential node as it has at least one detected event for all other products and its average edge weight is the highest by a margin of 5%.

Taking it further and using the Focus functionality on the three diesel nodes, for all of them the top 5 inbound products include NYHarborDiesel, GulfCoastDiesel, and LosAngelesDiesel. This means that even if in the graph some products appear to be out of their group, this might have happened because of the heuristics used to generate the tree or because some other product might really overpower expected behaviors.

Next the most glaring aspect of the graph is how influential specific products are, the tree isn't just an assorted web of relations but groups of products aggregating around very influential/influenced products. There are some expected trend setters like the OPECBasket that is used as a benchmark for oil price, the Gulf Coast refineries and then some unexpected like the Minnesota retail gas price.

Apart from the points already observed there are no other substantial conclusions expected like strong geographical links or chain influences like refinery price -> state price -> local price.

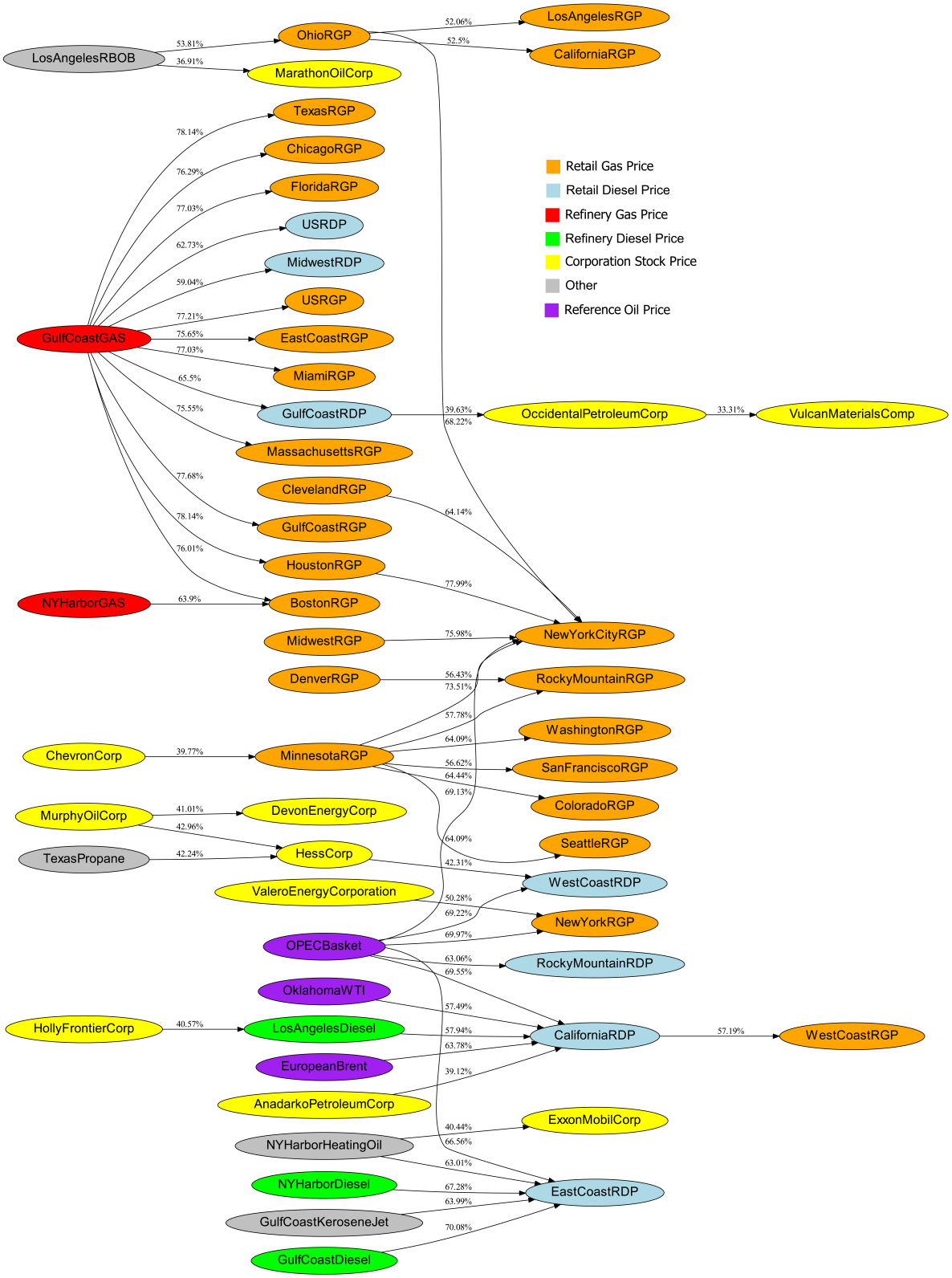


Figure 5.3: Graph showing the resulting Buy tree after applying Ramex Forum on the data with the parameters on Table 5.1.

While the final output of Ramex Forum is good for the “big picture” it uses an heuristic and the result displayed isn’t necessarily the best. On top of that, going from a graph to a tree, while easier to read, discards a lot of information.

The *Counters* functionality offsets this flaw and was used with a threshold set to 50% so that only edges heavier than that are considered, leading to only somewhat meaningful relations being included. Several things of note can be seen on the resulting table, first for influenced products, the top half of the table is populated exclusively by retail prices while the bottom, as expected, contains most refineries with no or very few inbound influences. Second, for influencing products, refinery prices accompanied by the OPEC Basket and the European Brent now dominate the top with a high average weight and edge count. However the national Oklahoma WTI doesn’t show much weight in influencing the USA’s prices, the same is true for all the included Corporations. Like in the tree in Figure 5.3 the power behind the Gulf Coast refineries and the OPEC Basket can be seen, as well as the New York Harbor refineries not previously visible.

So far only the connections themselves have been considered, if the influence weights are also taken into account the analysis becomes more complex. Edges with weights close to 50% have no certainty in indicating the behavior between the two products as it can neither tell us that one product consistently influences another nor that one product definitely doesn’t influence another. However it is still enough to indicate that there is some relation between nodes. For future work it might then be interesting to implement a "radius" filter around the 50% so that edges close to it by 5, 10, or X% are discarded this way edges will clearly either be “strongly related” or “strongly unrelated”.

With these points in mind the reading of the graph changes slightly, now with corporation stock prices being almost all connected to other products with weights close to 40% they can be considered outside of the sphere of influence of the oil and its byproducts and even not influential between themselves. This type of conclusion - that one product is outside the sphere of influence of another - might be very useful for building diverse stock portfolios. If the graph weight is minimized instead of maximized, the tool can reach groups of products that are independent of each other producing a quantifiably diverse portfolio.

For the Sell comparison type, an equal graph was done with the color coding and the results were very similar with strong groupings of colors and some few select products influencing groups of others.

In this case the GulfCoastGAS still shows as a key influencer but this time the retail diesel prices were replaced by refinery diesel prices, in other words, refinery gas prices have a big influence on refinery diesel prices when in a downward tendency.

This time influence over retail gas prices is concentrated around the NYHarborGas and there is a closer grouping between Corporations around the ValeroEnergyCorporation and between diesel prices around the oil benchmarks. Once again refinery diesel prices don’t seem to have a big influence over retail diesel prices.

A negative point in this second result is that most of the edge weights are close to the

50% mark, and as previously stated this isn't a good result in terms of usability of the information. This might be because of the usage of such a low moving average, while the edge weight average might be higher in the first stage of the algorithm, the value of the heaviest edges seems to be lower.

This first case study already provided very interesting results, some of expected chains of influences were found and clearly identifiable. While this is enough to show that the tool functions well for ideal cases, it is still unknown how it behaves in non-ideal situations. This will be handled in the next case study.

5.2 Case Study 2: Foreign Exchange Market

The exchange market was chosen for the second case study, this marketplace deals in the trading of monetary currencies. A countries' currency is valued on several different factors, from the gross domestic product, and expected future gains or losses to economical or political stability, on top of that the value of each currency is given in relation to another currency. This means that there is no constant base that the currencies can be compared to, one base currency must be chosen and the value of all other currencies is in relation to that base.

The exchange market is very volatile and dependent on too many things to consider them all, still international markets react with each other and with that so do the currencies. That leads to the expectation that there are at least some weak connections between them. Given that, in this case study there are no expected behaviors other than low values of influence, its purpose will then be more in line with "How will the algorithm react to a low influence scenario?".

5.2.1 Data and Information Gathered

Once again there was a need for specific data that isn't easily accessible, the free and reliable sources found were focused on specific themes and as such didn't provide the whole collection of currency exchange rates.

At the time of writing the best source found was the Federal Research Bank of St. Louis [Stl] that focuses on economical research. They provide datasets with the price of 18 currencies in relation to the U.S dollar and 4 values for the U.S. dollar in relation to other currencies. As for the reliability of the data, in the website from which the data was downloaded the following excerpt is displayed:
"The following exchange rates are certified by the Federal Reserve Bank of New York for customs purposes as required by section 522 of the amended Tariff Act of 1930. These rates are also those required by the SEC for the integrated disclosure system for foreign private issuers. The information is based on data collected by the Federal Reserve Bank of New York from a sample of market participants."

So while the not the complete list of available currencies, it's the ones considered most relevant by the Federal Reserve Bank of New York. Also, by Bloomberg's standards, all "Major" currencies are contained in the dataset.

The collected currencies were divided into the four geographic containers used by Bloomberg (Americas; Europe, Middle East, and Africa; Asia-Pacific) and the individual currencies can be seen in Table 5.2

Table 5.2: Defined parameterizations.

Americas	Europe, Middle East, and Africa	Asia-Pacific
Brazilian Real	Danish Kroner	Chinese Yuan
Canadian Dollar	Norwegian Kroner	Hong Kong Dollar
Mexican New Peso	Swedish Kronor	Indian Rupee
U.S. Dollar to Australian Dollar	Swiss Franc	Japanese Yen
U.S. Dollar to Euro	South African Rand	Malaysian Ringgit
U.S. Dollar to New Zealand Dollar		Singapore Dollar
U.S. Dollar to British Pound		South Korean Won
		Sri Lankan Rupee
		New Taiwan Dollar
		Thai Baht

5.2.2 Setup for the Case Study

As previously the data was normalized using a min/max normalization, its moving averages were calculated, and everything was uploaded to a database.

Using the observations of the first case study, the moving average size was set to 120 days and the threshold was kept at 1%. Because there are no expectations of influences, there is also no set period where they are anticipated, the value of δ will then be set on what is considered a reasonable size of 60 days, which translates to a period of 3 months (each month has four weeks of five working days).

The timeframe is set to 1999-2013 due for no particular reason other than it's a big enough time interval where all coins were already in use (the EURO only started being used as an accounting currency on January 1 1999).

In the first case study there was a known tendency for only price increases in oil being translated into changes in the pump prices, with price decreases not having so much effect. With the exchange market this isn't the case and as such its important to take into account both upward and downward trends in prices. The influence type parameter will then be set to *All* so that both *Buy* and *Sell* influences are captured in the same result.

5.2.3 Results

With all parameters set the tool was then used to produce an influence tree between the various currencies, the output can be seen in Figure 5.4. Once again, to facilitate the reading of the groupings, the tree was colored. This time the coloring was done according to the three groups seen in Table 5.2 with orange for Americas, blue for Europe, Middle East, and Africa and yellow for Asia-Pacific.

Once more the algorithm produced groups that are clearly identified by color, there is a single node that isn't connected to another node of the same color. However even in that case it can be justified that the way Bloomberg lumps Africa with Europe is not a good grouping, given that the only blue node not connected to another blue node is the African one.

If the result is to be trusted, the groups could probably be broken down into sub-groups to distinguish between the separate clusters of the same color.

Lending strength to the notion that currency values are more dependent of their country of origin than international markets, not a single influence has a weight above 50%. So in this case study there is no opportunity of saying that currency X frequently influences currency Y.

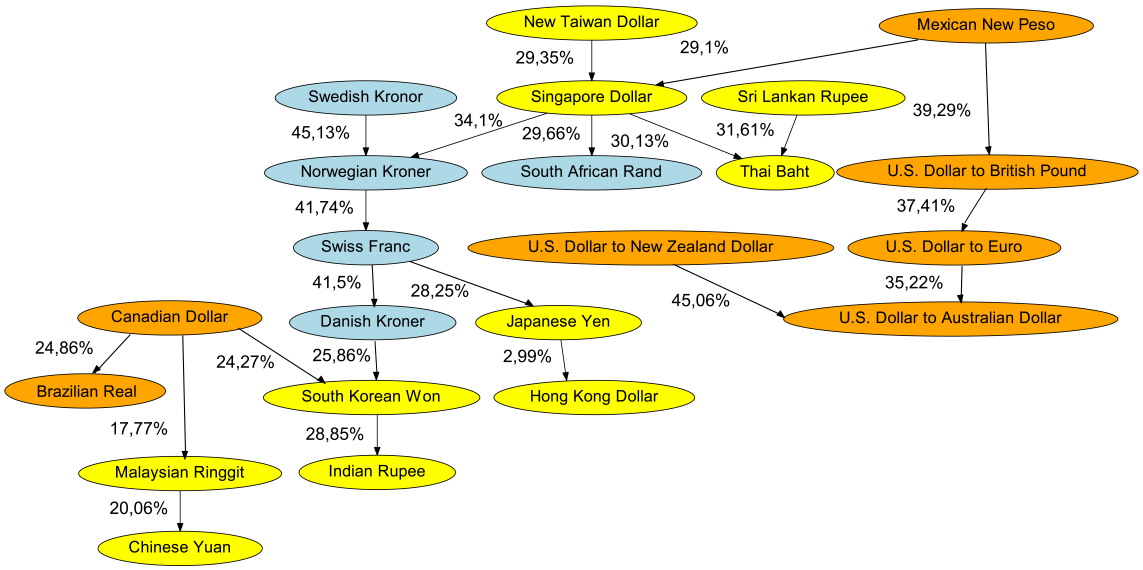


Figure 5.4: Graph showing the resulting All tree after applying Ramex Forum on the exchange market data.

In a final attempt to find if there is something else the tool can show, the Counters window was used. The only thing that stood out is how independent the Hong Kong Dollar is, it only has nine inbound connections with an average edge weight of 1%. Even among all the other low value influences, this one really shows how its value is completely outside the markets' group. However no literature that corroborated this discovery was

found.

This market is usually a target for high frequency trading (HFT) and a whole different set of relations might be found if the analysis is done on a scale of seconds and the parameterization is adjusted for such a study. This is currently outside of the scope of this thesis, but in the future it might be very interesting to see which patterns of influences show up in that trading scale.

While not as good as in the first case study, there were still interesting results. The edge weights were indeed low, but the relations found were still meaningful. Showing that even for groups of products that aren't heavily related, relevant connections are still found using this tool.

These first two case studies looked at small sets of products that were all clearly grouped for their relation to a certain market. Meaning that it is still unknown how the tool behaves with large sets of products from distinct markets of operation, this will be looked at in the third and final case study.

5.3 Case Study 3: Investment Fund and its Components

The third and final case study will focus on two investment funds, one that invests in over one hundred products and another with only around twenty five. This way there is a clear comparison of how the number of products in the input list influences the output.

Investment funds are products that work sort of like product portfolios managed by a third party. A fund has a manager that decides what products the investment fund invests in with all the money pooled from the clients that decided to buy part of that fund. There are several types of funds but the difference between them is usually only the underlying product type that is invested in.

This is a great situation for a case study as it covers some situations not addressed in the previous two cases, here there is a direct, known, and proven influence from a large group of products to a single one. There will also be a much larger pool of products that will probably also have influences between each other. As a final test, this case study will finally look at the stock market, one of the most interesting financial markets for the tool.

5.3.1 Hypothesis

Each fund is given a price depending on how well the invested products behave, meaning that there is a direct influence between the underlying products and the value of the fund.

At first glance this is a clear case where it would be expected to find lots of meaningful influences. This might be the case for the smaller fund, as it has fewer underlying products and each might carry enough weight to singlehandedly move the fund's rating to be detected as an event. However for the larger one, the capital is distributed over such a large number of products that individually each product has only a small sway in the

fund, this means that when events are detected it will almost be guaranteed that at the same time several products changed value in coordination.

5.3.2 Data and Information Gathered

Most investment funds keep the full information of their composition private, the general area of investment is public but exactly which products and how much capital is invested in each is not always available. For this case study contact was made with people that could provide the composition of funds and details for two of them were provided. The two funds are "BL Global Equities" and "Capital Gestion Multi Bond" and the composition provided was effective as of January 2014.

With the composition of the funds, all the available historical closing prices of the listed products were collected, parsed like before and inserted into the database. Not all of the products contained in the funds had their historical prices available for download, however these products were not obtainable because they were somewhat obscure, meaning that most of the important products were collected.

From this a further filtering was needed so that only products that were already listed at the starting date of the study are included. This narrowed down the product list to 113 elements and the full list can be seen in the appendix.

The full product list is available in the appendix, and it includes a lot of big names like Microsoft, Unilever, CocaCola, and Cisco along with more specialized and niche companies, creating a very heterogeneous input.

5.3.3 Setup for the Case Study

By the same logic used in the second case study, the parameters will be set to a δ of 60 days, and a threshold of 1%, and a moving average size of 120 days.

As for the studied time span, the first case study examined 7 years, the second 14 years, and now a much smaller time period of 3 years will be used. The starting date was set to January 1 2010 and the ending date to January 1 2013 because some of the products were only started much more recently, some only up to 2012, and this way they are all given an approximate weight. While it is possible to only study a one year time span, for a case study it doesn't show a real depiction of product's behavior. For example, in the last couple of year the markets have been recovering and are currently in an almost constant upward trend without significant variations, as such the results will seem very positive because the studied period doesn't have much variety. The three year time span won't capture the 2007-2009 crisis but it still has enough small variations to provide, at least somewhat, relevant results.

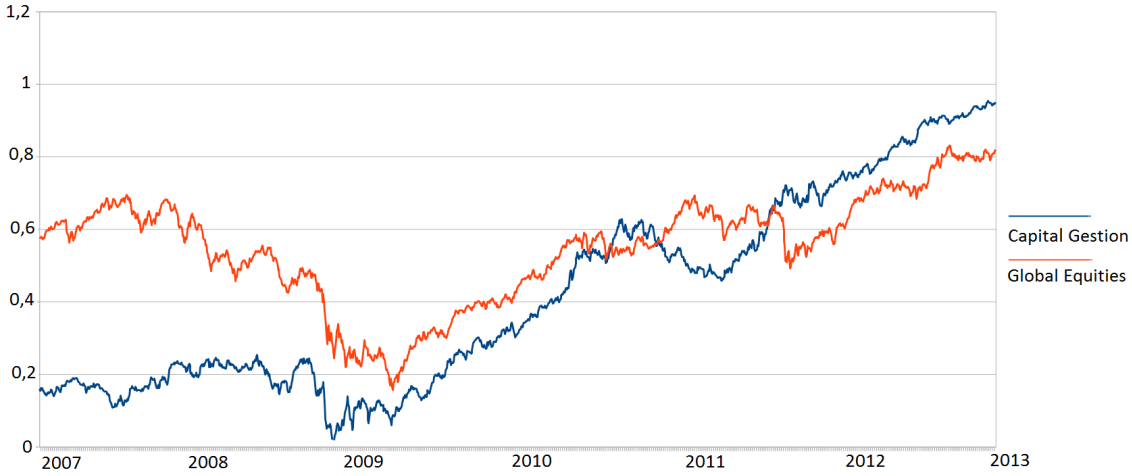


Figure 5.5: Fund value evolution along the 2007-2013 period.

While the 2010-2013 period has an almost constant positive trend, it can be seen that both funds have some significant dips in value around the start of 2011 that are enough to provide cases of *Sell* events.

5.3.4 Results

This time there are no preconceived groups that products belong to and as such coloring the nodes is not possible. That fact grouped with the larger amount of nodes makes for a harder to read graph. That shows that even though the Ramex Forum algorithm coupled with the tool do a good job at showing information in a more concise manner, depending on the objective, it might be important to choose the input.

For example, in this case study there is no objective other than “lets see what shows up” and as such throwing everything in the mix will allow some interesting things to pop-up. They will however probably be buried under a lot of irrelevant noise.

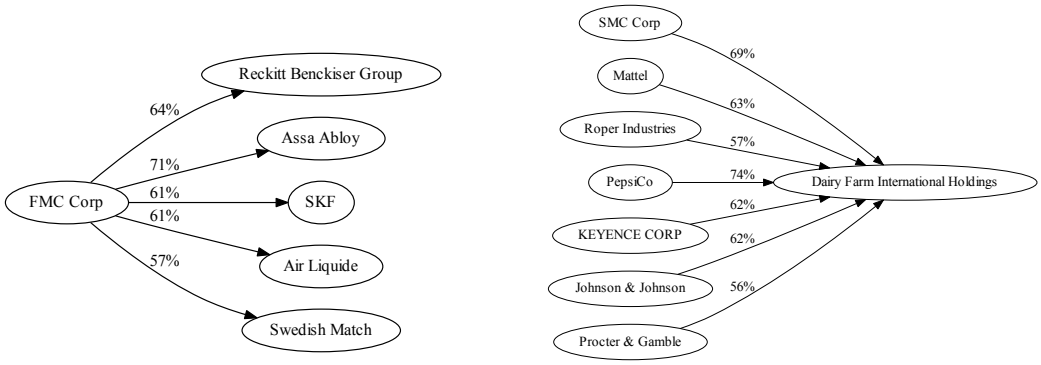
5.3.4.1 Buy

What initially stood out the most was that the product "Nederland 12" had two outgoing edges with a weight of 100%, this raised some suspicions and upon further analysis, with the edge focus window, the reason for this was clear. This product only started being quoted in 2012 and with a moving average of 120 days, in 220 working days per year, that only leaves 100 days to be analyzed. Of those 100 days only 5 where above the upper threshold and coincidently all of them count as an influence over another product, thus leading to the misleading value of 100%. Because of this it was decided that all products that could be identified as being quoted only after the start of 2012 would be removed. Those products are: America Movil 12, Deutschland 12-17, European Financial Stability Facility 13, IBRD 13, and Nederland 12.

With those products removed, 106 remained and the resulting graph had and average edge weight of about 56%. The highest edge weight is 75% so while there are no "certain" products, there are significant relations. Because of the quantity of products, it's impossible to show a readable graph in these pages, a written description of the most relevant products and their connections will then be made. The full graph can be seen in the appendix.

In this result there are four products that stand out for their number of connections, they are FMC Corp, Syngenta, SMC Corp, and Dairy Farm International Holdings. The first three stand out for having the biggest number of the heaviest outgoing edges (5, 8, 14) and the last one for the most incoming edges (7). In Figure 5.6 & 5.7 each sub-group is displayed with all their connections, identifying the products and the weight associated with the influence.

A brief description of each company's area of operation will be taken from Wikipedia and those will be used to try to find possible justifications for each influence. A small and concise evaluation of each relation will be done to help situate the reader.



(a) FMC Corp outgoing relations.

(b) Dairy Farm International Holdings incoming relations.

Figure 5.6: Focus of four of the most relevant groups of relations found.

FMC Corporation is a chemical manufacturing company based in America, it's mentioned that it also operates in the manufacture of pumps for various uses and pesticides for agriculture. It's expected that the products influenced by it are heavily dependent on chemicals or pump related works:

- (71%) "The Assa Abloy Group is a Swedish lock manufacturer, and is the world's largest lock manufacturer by sales volume." - strangely, the heaviest influence is towards a lock manufacturer. At least only looking at the relation by area of operation there is no clear reason for this influence.
- (64%) "Reckitt Benckiser is an English consumer goods company, producer of health, hygiene and home products." - health, hygiene, and home products are chemical dependent so this influence is justified.
- (61%) "SKF, is a Swedish bearing company, supplying bearings, seals, lubrication and lubrication systems, maintenance products, mechatronics products, power transmission products and related services globally." - while FMC Corp produces pumps for various uses, this relation seems inverted since SFK produces the materials needed for pump production.
- (61%) "Air Liquide, is a French multinational company which supplies industrial gases and services to various industries including medical, chemical and electronic manufacturers." - once again the relation seems inverted, it states that Air Liquide produces materials for chemical manufacturers.
- (57%) "Swedish Match is a Swedish company based in Stockholm that makes snus, moist snuff, and chewing tobacco" - a tobacco producer will certainly need some chemicals but that need probably doesn't have a very big influence on the companies' value.

Observations: Some interesting dependencies were found, but also, at first glance, there isn't much justification for some other influences. Further research by a market specialist is needed before we can know if they are either pure coincidence, or if there are some other more complex underlying factors.

Dairy Farm International Holdings is an Asian retail company and a major pan-Asian retailer involved in the processing and wholesaling of food and personal hygiene products in the Pacific region and in China. In this case it would be expected that this company would be somewhat self-sufficient, as it doesn't produce anything and it's value comes from how much it sells to the public, economical health would be the major influence in its value.

- (74%) "PepsiCo Inc. is an American food and beverage corporation, with interests in the manufacturing, marketing and distribution of grain-based snack foods, beverages, and other products." - while this company produces a wide range of products that are sold in retail sellers, it's not clear how its value would influence the retail company that sells its products.

- (69%) "SMC Corporation is a Japanese which specializes in pneumatic control engineering to support industrial automation." - no direct relation seems to exist.

- (63%) "Mattel, Inc. is an American toy manufacturing company." - same as PepsiCo, produces products that are sold in retail sellers.

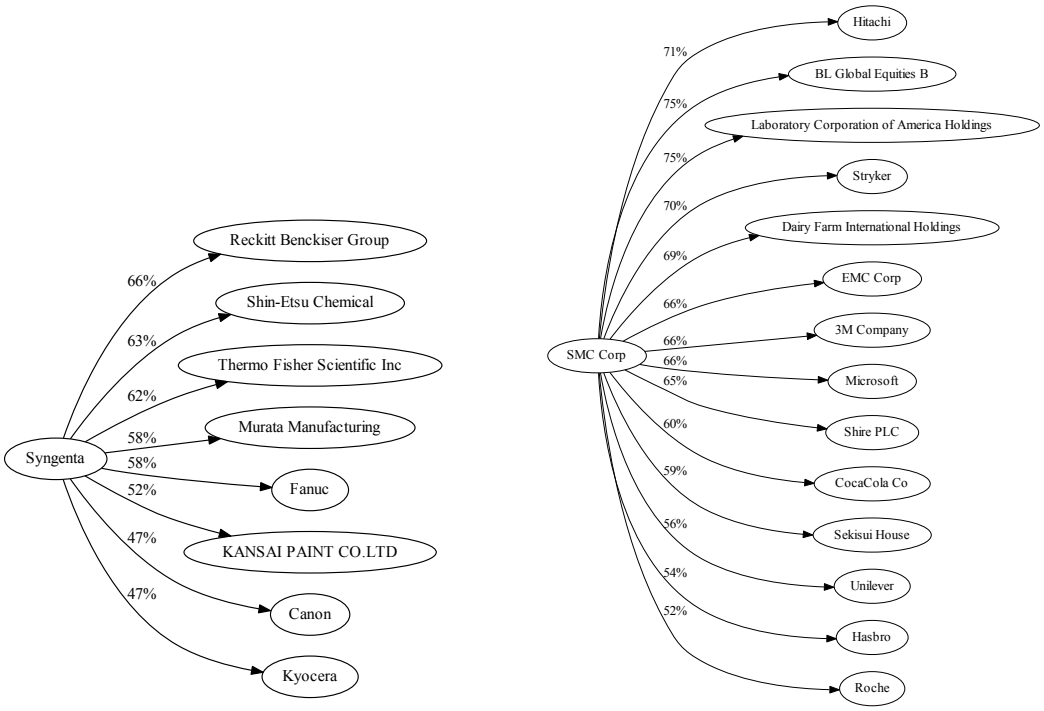
- (62%) "Keyence Corporation is a direct sales organization that develops and manufactures automation sensors, vision systems, barcode readers, laser markers, measuring instruments, and digital microscopes." - some of the products produced by this company are used in shops.

- (62%) "Johnson & Johnson is an American medical devices, pharmaceutical and consumer packaged goods manufacturer." - same as PepsiCo and Mattel, produces products that are sold in retail sellers.

- (57%) "Roper provides a wide range of products, the company has four main business lines: Industrial Technology, Radio Frequency Technology, Scientific and Industrial Imaging, and Energy Systems and Controls." - no direct relation seems to exist.

- (56%) "Procter & Gamble Co. is an American consumer goods company, its products include pet foods, cleaning agents, and personal care products." - same as PepsiCo, Mattel, and Johnson & Johnson, produces products that are sold in retail sellers.

Observations: from this case retail sellers seem influenced by the value of the companies that manufacture the products sold at the stores. This behavior goes in line with industry sectors, the secondary sector (product processing) manufactures the products that are sold in the tertiary sector (retail sellers).



(a) Syngenta outgoing relations.

(b) SMC Corp outgoing relations.

Figure 5.7: Focus of four of the most relevant groups of relations found.

Syngenta is a Swiss agribusiness that markets seeds and agrochemicals, it is also involved in biotechnology and genomic research. It is then expected that it influences businesses associated to agriculture or biotech related areas:

- (66%) "Reckitt Benckiser is an English consumer goods company, producer of health, hygiene and home products." - health, hygiene, and home products are biotech related dependent but this influence doesn't seem very solid.
- (63%) "Shin-Etsu Chemical is the largest chemical company in Japan, it has the largest global market share for polyvinyl chloride, semiconductor silicon, and photomask substrates." - while not exactly related by products, they are both big companies of a similar sector and that doesn't make this influence unexpected.
- (62%) "Thermo Fisher Scientific is an American biotechnology product development company, one of the leading companies in the genetic testing and precision laboratory equipment markets." - same as Shin-Etsu Chemical.
- (58%) "Murata Manufacturing is a Japanese manufacturer primarily involved in the manufacturing of ceramic passive electronic components, primarily capacitors, and it has an overwhelming share worldwide in ceramic filters, high-frequency parts, and sensors." - no direct relation seems to exist.

- (58%) "FANUC is a group of companies, principally FANUC Corporation of Japan, Fanuc America Corporation of USA, and FANUC Robotics Europe, that provide automation products and services such as robotics and computer numerical control systems." - no direct relation seems to exist.

- (52%) "Kansai Paints is the largest industrial paint and second largest decorative paint company based in Mumbai. It is a subsidiary of Kansai nerolac paints, JAPAN. It is engaged in the industrial, automotive and powder coating business. It develops and supplies paint systems used on the finishing lines of electrical components, cycle, material handling equipment, bus bodies, containers and furniture industries." - no direct relation seems to exist but Japan keeps coming up.

- (47%) "Canon Inc. is a Japanese corporation specialized in the manufacture of imaging and optical products, including cameras, camcorders, photocopiers, steppers, computer printers and medical equipment." - once again no direct relation seems to exist and its a Japanese company.

- (47%) "Kyocera Corporation is a multinational electronics and ceramics manufacturer headquartered in Kyoto, Japan. It manufactures industrial ceramics, solar power generating systems, telecommunications equipment, office document imaging equipment, electronic components, semiconductor packages, cutting tools, and components for medical and dental implant systems." - once again no direct relation seems to exist and its a Japanese company.

Observations: in this case it seems that the company itself doesn't have much relation to its influenced products but there is a large similarity between influenced companies' field of operations (ceramics and technology) and geography (Japan). It seems that this group showed up not only because of their relation to Syngenta but for similar qualities among themselves.

SMC Corporation is a Japanese company which specializes in pneumatic control engineering to support industrial automation. Industrial automation adds value to a company by allowing for faster and better production lines, with this in mind the logic behind SMC Corp being an influential company seems to be: a company contracts SMC to automatize their industry (the value of SMC goes up with the new contract), and news of this increase in productivity leads to an increase in value of the contracting company. This because the increase in value needs to happen withing 2 months because of the set δ. It will then be expected that the influenced companies rely heavily on industrial production:

- (75%) "BL Global Equities B" - one of the analyzed funds, it seems SMC Corp has a heavy influence on the fund.

- (75%) "Laboratory Corporation of America Holdings, more commonly known as LabCorp, is an American company that operates one of the largest clinical laboratory networks in the world." - maybe clinical labs require a lot of automation.
- (71%) "Hitachi, Ltd. is a Japanese multinational engineering and electronics conglomerate company, a highly diversified company that operates eleven business segments: Information & Telecommunication Systems, Social Infrastructure, High Functional Materials & Components, Financial Services, Power Systems, Electronic Systems & Equipment, Automotive Systems, Railway & Urban Systems, Digital Media & Consumer Products, Construction Machinery and Other Components & Systems." - all of these segments have strong industrial components.
- (70%) "Stryker Corporation is a medical technologies firm, their products include implants used in joint replacement and trauma surgeries; surgical equipment and surgical navigation systems; endoscopic and communications systems; patient handling and emergency medical equipment; neurosurgical, neurovascular and spinal devices; as well as other medical device products used in a variety of medical specialties." - mass production of equipment requires a large industry
- (69%) "Dairy Farm International Holdings" - already discussed, doesn't make sense that a retail company requires much automation.
- (66%) "EMC Corporation is an American corporation that offers data storage, information security, virtualization, analytics, cloud computing and other products and services that enable businesses to store, manage, protect, and analyze data." - not related in any way to industrial automation, no direct relation seems to exist.
- (66%) "The 3M Company, formerly known as the Minnesota Mining and Manufacturing Company, is an American conglomerate corporation that produces more than 55,000 products, including: adhesives, abrasives, laminates, passive fire protection, dental products, electronic materials, medical products, car-care products, electronic circuits, and optical films." - massive production industry that benefits greatly from automation.
- (66%) "Microsoft Corporation is an American corporation that develops, manufactures, licenses, supports and sells computer software, consumer electronics and personal computers and services." - while mainly known for their software, there is also a significant portion of their business that focuses on hardware, the production of which requires a production industry.
- (65%) "Shire Plc is Irish-headquartered global specialty biopharmaceutical company that develops and provides health care in the areas of behavioral health, gastrointestinal conditions, rare diseases, and regenerative medicine." - the production of pharmaceuticals requires an industry.

- (60%) "The Coca-Cola Company is an American multinational beverage corporation and manufacturer, retailer and marketer of nonalcoholic beverage concentrates and syrups" - the manufacture portion is industrial based.
- (59%) "Sekisui House is one of Japan's largest homebuilders." - home building is not exactly industry based. However the description is not very informative, maybe they also manufacture some of the products used in construction.
- (56%) "Unilever is an Anglo–Dutch multinational consumer goods company, its products include food, beverages, cleaning agents and personal care products." - Unilever doesn't partake in the manufacture of products but owns the companies that do. No direct relation seems to exist.
- (54%) "Hasbro Inc. is an American toy and board game company. It is one of the largest toy makers in the world." - toy production is industry based.
- (52%) "F. Hoffmann-La Roche AG is a Swiss health-care company that operates worldwide under two divisions: Pharmaceuticals and Diagnostics." - same as Shire Plc.

Observations: while it is true many of these influenced companies are industry based, and that they could benefit a lot from industrial automation, the premise for the influences seems too far-fetched. The same goes for the influence between SMC Corp and the fund, at 75% it seems too high to be a true influence of a fund that is composed of almost a hundred products. It is then concluded that while SMC Corp does rise in value before several other companies, at the surface there doesn't seem to be any real link between company values.

Final observations

Contrary to what was expected, and seen in the second case study, for stock and fund prices there is no clear sequence of primary sector (resource collection) to secondary sector (processing) to tertiary (delivery and sale), except for Dairy Farm International Holdings where the is a strong secondary to tertiary pattern.

The four examples analyzed here are not a representation of the whole result obtained, however from such a small sample there where very interesting results, from no apparent relation whatsoever to clear industrial sector sequence. This small and not very through analysis shows that, even by just taking random picks from the results obtained with Ramex Forum, justified patterns and relations are found.

Using the other tool functionalities, further information can be obtained. Setting a minimum edge weight of 50%, the two most influential products are SMC Corp (as previously seen) and JSR Corp (Japan Synthetic Rubber Corporation, no Wikipedia page) that for some reason didn't stand out in the final tree. Both products influence others (34 and 25 respectively) with an average edge weight of 59% and their heaviest influence is

75% (on different products). As for the most influenced, Hitachi, Shin-Etsu, and Dairy Farm International Holdings all have an average edge weight of 58% from 22, 21,and 19 products and their heaviest influences are 73, 70, and 74 respectively.

Finally, both funds' values don't appear to be very influenced by their respective products, as expected.

5.3.4.2 Sell

As for the *Sell* results, only a small comparison to what was seen for *Buy* will be done. First of all the average edge weight is lower by 11% at 45%. Second, none of the big influence groups seen previously are present, now the most influential products are PepsiCo (still influencing Dairy Farm International Holdings), Schneider Electric, Microsoft, and JSR Corp. As for influenced products, there are none that contain five or more inbound edges.

The result seems loose with very few groups and that causes mostly uninteresting relations, this is most likely caused by the positive market trend along most of the studied period. However the case might be something entirely different, in [Har+10] it's shown that market segments fall in cascading patterns, that is, when enough actives fall in value the markets tend to follow as one. At first glance it might seem that what this leads to is the identification of those products that fall first. The problem is that it isn't always the same ones that fall first, and as such the final result will be spread over a large number of products that don't influence anything, but were just a catalyst for a market fall at one point. Which in turn leads to the inconclusive results obtained in the *Sell* results.

5.3.4.3 Counter-Cycle

All *Counter-Cycle* results obtained so far have been very weak, while it is expected that products shouldn't go against the market, it seems that there is almost no exception to this. For this case the average edge weight is at 30% and of all the cases that stood out (more edges or higher weights) when studied in further detail it was just that one of the products had a lot of variations while another stood stable for longer, causing what the algorithm detected as counter-cycles. Even when true counter-cycles occurred, they where mostly isolated cases even within that products' history.

5.3.5 Simulator Results

Using the simulator previously discussed in Section 3.7, a simulation was done with the data used for this case study. The analysis was done over a period of 5,5 years starting in 2008 and ending in mid 2013, capturing the end of the financial crisis and the start of its recovery at the start of 2009. It took a while to get the parameterization right, initial simulations fared barely better than the control and at times held on to products that caused major losses. But with the correct minimum support(usually three or more), various thresholds(minimum confidence of 50%, 5% threshold from the moving average), moving average size(120 days), and δ(30 days) the results look very promising.

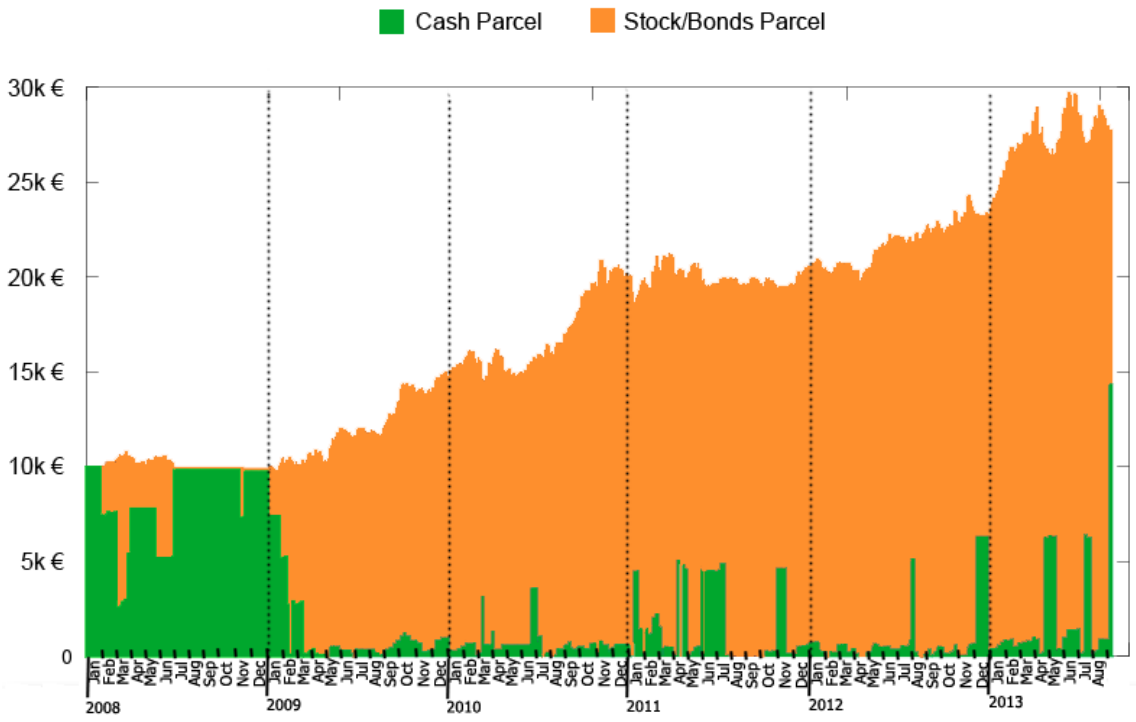


Figure 5.8: Division of the wallet between cash (green) and products (orange).

A first reaction might be that the parameters where optimized for this case and won't work as well in other cases. As a rebuttal to that, five and a half years have a lot of different scenarios that would make it hard to fit the parameters to this whole period. Nevertheless, more extensive tests need to be done before the application of this simulator to investing decisions. Still, the simulator is already to enough to validate the results obtained with the tool.

Using all of the graphs together provides a bigger picture that helps understand how the market, and consequently also the algorithm, behaved. For example, looking at the Figure 5.8 there is a period around the 200-400 days where there are no products in the wallet, this is clearly explained by looking at Figure 5.9. In that period it's obvious that the market (green line) is in clear decline, making it a bad moment to invest. As the market starts to pick back up again closer to the 400 day mark, more and more products begin rising above their moving averages and buy signals start to be given. With that in mind it makes sense that the algorithm wouldn't have any buy signals active for that period.

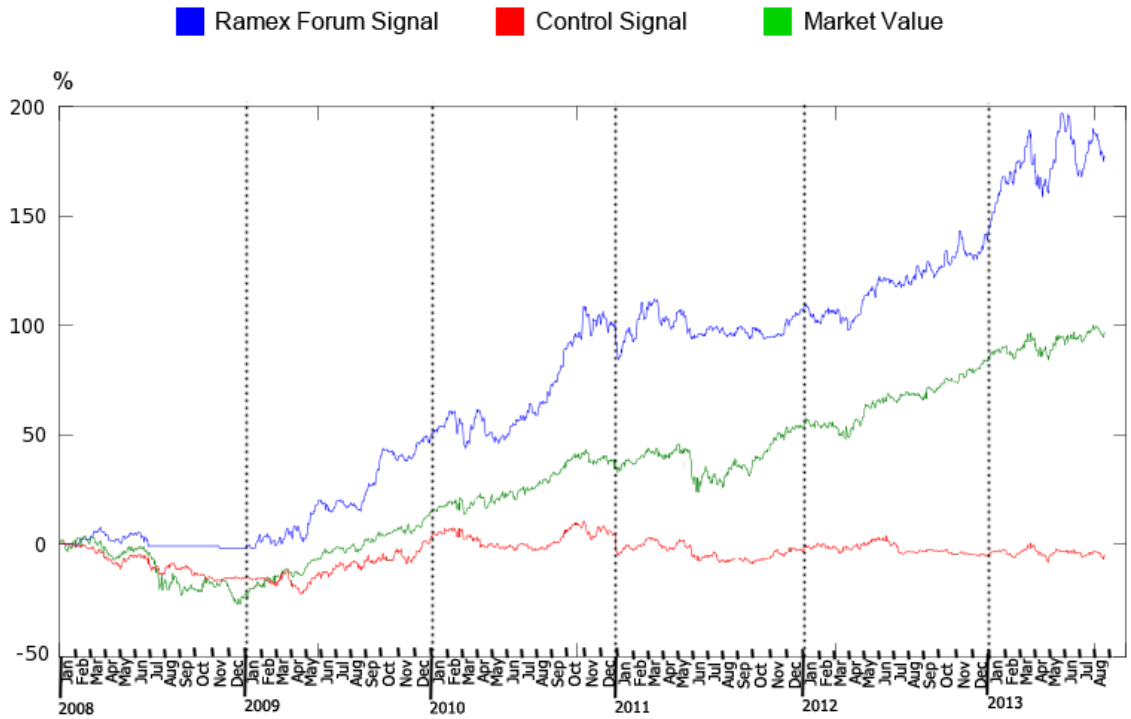


Figure 5.9: Percentual value change, in relation to the first day of simulation, of the simulated wallet, control wallet, and market value.

By comparing the market line in Figure 5.9 with the fund values in Figure 5.5 in Section 5.3.3 the reader can see that the market value is very close to the sum of the value of both funds, as expected. Even if the input doesn't contain all the products that make up these funds, the most significant ones are in enough number to provide a value very close to reality. This helps validate the results obtained in this case study.

Both Figure 5.8 and Figure 5.8 show that there were significant gains along the studied years. At the end of the five and a half year period the wallet increased in value by close to 200%, this is about a 3% gain in value per month, or 36.4% per year which is a great result. However there is one thing that the simulator currently doesn't take into account, and that is commissions.

Usually stocks and bonds are bought through a broker that charges commissions between 0.5% and 5% depending on the volume of trade. This puts a massive dent in the profits obtained by the simulator. Because it tries to buy and sell products as often as possible to avoid dips in value, it would start to rack up a lot of costs in commissions. However, looking at Figure 5.10 it's clear that the algorithm is giving mostly good signals, the average percentage for correct buy signals is close to 98%, or about 35% more than the control that hovers closer to the 65% mark.

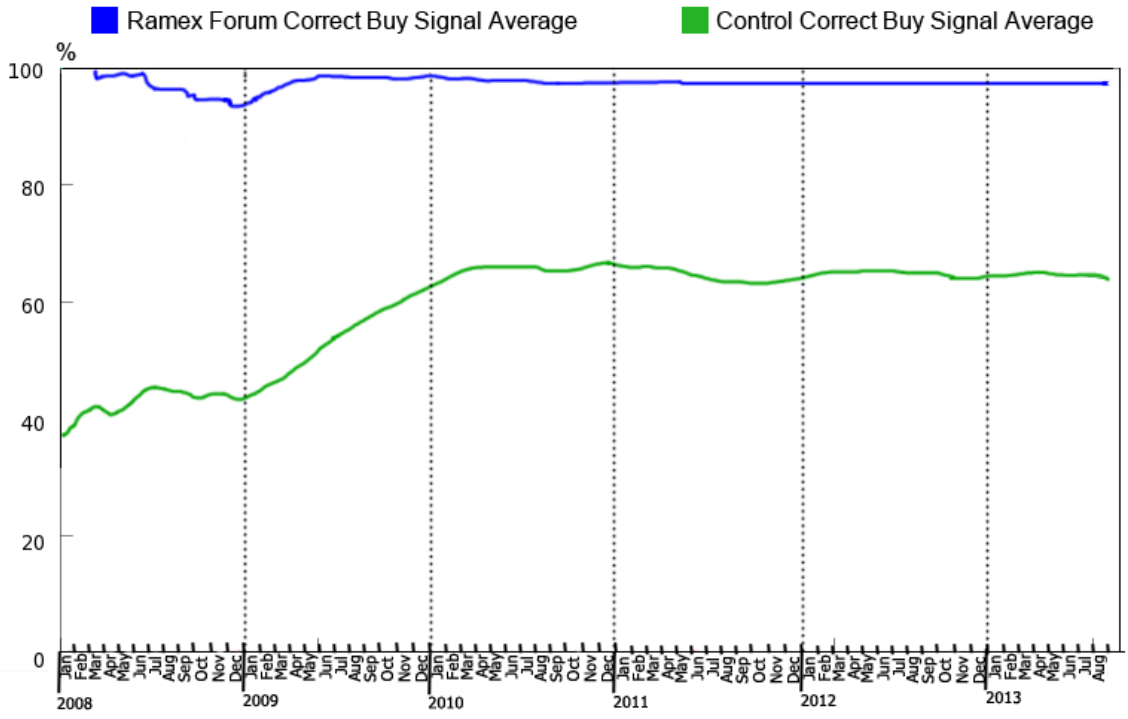


Figure 5.10: Comparison between the percentage of correct buy signals from Ramex Forum and the control.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

In the beginning of this thesis four criteria were set for the tool as objectives, and three points as contributions of the thesis. Looking at the results obtained in the three case studies it can be empirically argued they showed that all of them were successfully met.

The Ramex Forum algorithm was successfully improved and adapted into a tool that verified it can find relevant sequential patterns, in a way that is clear enough to even allow that discovery of micro and macro patterns in financial data. The tool was built around the output of this algorithm in a way to provided several visualizations of the data that, allowing different views of the same information, provide a deeper understanding of the analyzed data.

The three case studies were done with distinct sets of data, in a carefully studied parameterization, and varying lengths of analyzed periods. The results of these case studies were very interesting for *Buy* influences and somewhat less for *Sell* and *Counter-Cycle*, but it was still enough to validate the use of this tool and justify further developments to see how useful it is in the real world.

As the thesis was developed, some possible improvements over the algorithm and the tool were detected. The improvements that were important for the results and deemed feasible in the available time-frame were developed and implemented. Other identified developments that are either too complex to develop in time to the thesis, or weren't considered relevant enough for it, will be mentioned in the section Future Work.

6.1 Conclusions

6.1.1 Case Study 1: Petroleum and its Derivatives

The first case study was quite useful for a starting demonstration of the usage and capabilities of the algorithm and tool. It demonstrated the algorithms' ability to find patterns that

further translated into being capable of grouping together similar products due to their interconnectivity. On top of that the usefulness of such a tool starts to look promising, from building diverse portfolios with proof of independence between products or knowing that a product might increase or decrease in value with upwards of 70% certainty is very interesting.

As for the hypothesis set before the case study (that there should be a chain of influences from price of crude oil to refinery prices to retail prices), while not completely corroborated, it was still found that the market for oil and its derivatives follows part of the expected chain of value. Changes in the price of oil are not as influential in refinery prices as expected, however refinery prices are very influential in retail prices. On top of that there is no immediately discernible geographic correlation between price changes.

Finally, the value of corporation that deal in the collection, refinement, and distribution of oil products is somewhat outside the scope of influence of oil trading itself, at least there were not enough detected sequential changes in prices between the corporations' value and oil derivatives prices. Even between themselves the number of detected events leans more towards the "not related".

6.1.2 Case Study 2: Foreign Exchange Market

Given the premises, the lack of anything new from the second case study is not unexpected. The number of products is very low and the value of each product is mainly derived from things not contained in the input.

However as an important point, it shows that the algorithm doesn't produce false positives. This means that if a large list of products of which their inter-relations are unknown, the output will probably be separated into two main groups of "Connected" and "Disconnected" products. With the first group containing all the nodes connected with heavy edges and the second containing the rest, or if there are no connected products, just one group with sub-50% edges. Still, even in the disconnected group the relations might still be very relevant and should also be considered.

6.1.3 Case Study 3: Investment Fund and its Components

The last case study came much closer to how the tool is be expected to be used in a real life situation. The results obtained for *Buy* while not ground-breaking, still managed to raise some surprises and interesting connections. As for *Sell* and *Counter-Cycle* the results obtained were pretty much uninteresting, mostly due to the situation in the studied time period.

As for the tool, with that number of products the graph starts to become harder to navigate, finding a specific node becomes hard and a zoom functionality seems a necessity. The chart functionality was a great help in justifying the problems with the *Sell* and *Counter-Cycle* results. The counters table while useful to find the biggest influencer and

influencee, raised some some difficulties in finding interesting outliers because of the number of products.

Still, what the results showed the most is that the influences found don't seem to be true influences, but that some products react to changes in specific sectors faster than others, thus creating the "influences" found. Most importantly is that connections were found and the results aren't just trash.

The small simulation done with the products of the final case study, about 100 of them, managed to yield some very interesting results. Even if the final gains aren't a direct relation to how the usage of these signals would work in reality, it still shows that the signals are very reliable with a correct average of about 98% seen in Figure 5.10. With more products, more influences would be found and the minimum support could be increased even further, hopefully providing even more (both in quantity and quality) accurate results.

At this stage of development no one would trust on the signals provided for a completely automated system, however coupling the experience of an economics analyst with a list of products that are possibly going to increase or decrease in value soon looks very promising. Further work still needs to be done on this simulator and the analysis of its results, as stated before a further focus on selling products needs to be done to reduce losses. There is still a lot of information on the graphs that isn't completely analyzed, for example comparing gains per month to see if the signals are always positive or if there are some months that the performance is worse than the control and market, and if so why.

Plans are being made to implement an extended version of this simulator in a small financial boutique so that simulated client portfolios can be managed by analysts in real timed based on signals given by the algorithm. With this a clear evaluation of real use can be made by experts in the area.

6.1.4 Final Conclusions

In regards to the objectives of this thesis, the parameterizations were successfully studied and applied for the desired framework. The full analysis was done in Section 5.1.3 and it was the sensibility of the results, in relation to the parameterizations, that allowed the choice of which parameterizations should be selected for a dependency analysis of up to 60 day periods.

The testing of the tool and algorithm was done both in terms of synthetic tests (Sections 3.6.1 & 3.6.2) and in real world data (Case studies 1, 2, and 3). These showed that the algorithm works as needed, and that the tool successfully makes use of Ramex Forum to generate useful results.

By allowing the discovery of macro patterns, the visualizations of information included in the tool showed that they were good ways to visualize the sequential patterns, as required. All four visualization modes seen in Chapter 4 had some part in the analysis of each case study, meaning that they were all good ways to visualize the information

generated by Ramex Forum.

The developed tool turned out quite a success, besides achieving all the goals set for its development the results obtained proved that it is useful. Relevant sequential patterns were found and usable results were achieved, leading to interest in its full development by sector professionals.

The Ramex Forum algorithm was successfully improved, analyzed, and tested creating a solid academic basis for its behavior and output. Meaning that future works can build upon this thesis with full information on how it is implemented, what its temporal and spacial complexities are, and how the output can be managed to be efficiently displayed.

6.2 Future Work

6.2.1 Interval-Based Mining

In Section 2.4 Allen's Relations were introduced, these definitions allow a more complex understanding of how the products relate to each other. The Ramex Forum algorithm only looks for one relation type (overlaps) of the 7 presented, by taking into account these other informations other types of influences and interactions between products could be detected.

Implementing the detection of these events wouldn't be very complex since the way consecutive event days are kept already allows for the detection of all of the new events except for "before" and "meets". This however wouldn't be the complex part, other points need to be addressed first such as how would these new relations affect how an influence is considered? Would they be helpful in detecting patterns in financial markets or would they just complicate things?

Further analysis needs to be done in order to conclude whether this addition is worth the effort.

6.2.2 Distributed Database and Computing

One of the most time-costly steps is retrieving the data from the database, on top of that the algorithm completely stops when waiting for the data of the next moment. This problem could be lessened in two steps, first by increasing the speed at which data is retrieved by the database and second by reducing CPU downtime.

While not directly a change to the algorithm, the database schema and implementation is part of the tool. By creating a schema and database structure that supports the distribution of the information, the time to retrieve data can be reduced.

As for reducing CPU downtime the first step would be to start retrieving the data for the next moment as soon as the first one is retrieved and not after the computations are done. This way the CPU and database access will be concurrently busy for as long as the processing takes.

Making further use of the distributed database, the computations could also be distributed given that each day can be processed individually and then the final output can be obtained by looking at the individual result for each day.

These improvements would be more useful for much larger computations than the ones done for this thesis and as such no actual study of how it could be accomplished was done.

6.2.3 Graph Interactivity

The current tool produces static and non-interactive graphs, for small and medium sized ones this isn't much of a problem as it can usually be seen whole and it's easy to keep track of where is what. As for large graphs it would be useful to be able to move single or groups of nodes to organize the graph in a more readable way.

Being able to color the nodes inside the tool would also be useful, it's clear in the case studies that coloring nodes according to some categorizations can help in visualizing groups. Doing this inside the tool would be another step in creating better results.

One more thing that would provide a more intuitive interaction with the tool is being able to click nodes and having the information related to that node immediately displayed. Pulling up the focus window and searching through the whole list of nodes to find the one that is needed could greatly be simplified and expedited with this functionality.

Graphviz already provides functionality to easily color nodes and this could be achieved without the need to change much in the tool. Supporting interactive would need a completely new way of generating nodes so that their location could be identified and changed on-the-fly.

6.2.4 Influence Uncertainty Filter

As expected and seen in the case studies, some influences will have weights close and around the 50% mark. An influence with that weight can be interpreted as "Half the times A goes above/below the threshold B will react." and predictions based on this result would be as reliable as a simple coin toss.

If the tool is being used to find relations that are positive predictions of product behaviors and not just to find relations between them, it would be useful to exclude these uncertain influences. This can easily be done just by creating a filter that removes edges that are close to the 50% value by a certain amount.

The reason why this wasn't implemented in this thesis is that it requires further extensive study. There is a need to evaluate if the value should always be 50%, in periods where the market is very skewed to either side, this value might need to be increased or decreased. Furthermore, an in-depth evaluation of the implementation and results would need to be done, and that was too much work for the contribution this filter is predicted to provide.

6.2.5 Minimum Influence Tree

Currently the Ramex Forum algorithm is only able of finding maximum (or close to) spanning trees, this means that it is looking to find the strongest connections between products. As stated in the second case study, for some cases it might be interesting to find the minimum spanning tree, thus generating a minimum influence tree.

The most immediate use for this functionality would be to find independent stock products, when grouped together these could produce a portfolio of products that are independent between themselves. Diverse portfolios are useful for lower risk investments since the products do not react to each other or in a similar fashion, and its expected that they won't depreciate in value all at once. This way the portfolio "protects" itself by always having some products that will keep the value of the whole portfolio from falling too much. However this will also have the opposite effect of keeping the portfolio value from rising too much, thus leading to a portfolio with a steady value, hopefully with an upward trend.

6.2.6 Product Aggregation

Based on the third case study, it already becomes clear that as the number of products go up, the readability of the graph falls. It might then be interesting to allow the aggregation of products into a single node. This might be very interesting in cases where there is a need to find relations between specific market sectors.

If all the products are aggregated into a single value that represents all of them (an average of all values or something equivalent), the number of nodes can be greatly reduced as a large group of nodes is compared at once. This will be even more interesting as a way to find macro-patterns.

6.2.7 Pre-calculated State Database

For each product in each moment two values need to be retrieved from the database and then calculations need to be done over those values to evaluate whether in that day the product is above, below, or between thresholds. This might not seem much, but multiplied by the number of products and then all the analyzed moments means there is a lot of time spent over these calculations. Furthermore while using the tool it's usual to run the same period of data with small changes in parameterization, that will lead to the same calculations being done several times.

By creating a database of states, whenever the state of a product is calculated the first time, it can be immediately inserted into the states database. Reducing the time needed to create the graph in the future by only requiring one database access per product and no calculations.

However this might generate a lot of disk usage overhead, each state will need to be tied to the threshold, product type, and moving average parameterizations because changing those parameters leads to different states. It will then be necessary to have a way

to manage what information is kept, either pre or pos generation of said data. A good way to deal with this is to identify which states might be necessary in future runs or to justify obtained results on later audits, and only keep that data.

One other problem with this is how to identify if there is a state entry on the database. Either a query is done before the normal procedure, meaning that one extra query is added for every calculation that hasn't already been done, or some sort of bitmap table indicating if the state has been calculated or not is needed.

One way to reduce the performance hit that comes with the first choice is to give the user the option of activating the states database or not. If the user knows this is the first time he is running that particular period with those parameters, then he can disable the functionality and suffer no performance gain or loss.

This has been partially implemented for the simulator but instead of the states being inserted into the database as they are calculated, an SQL script was created that calculates these values for all the entries in the database. While being very time-costly, it is a one time thing that proved to significantly reduce the time spent doing test runs.

6.2.8 Advanced Simulator

The small simulator created was of great use to have an idea on how the signals given by Ramex Forum can help in building a profitable portfolio. There were still some small details that still kept it from being an exact simulation of reality, for example not taking into account the commissions charged when buying stocks.

There is still some room for improvement of this simulator such as multiple portfolios, better loss prevention, and monthly evaluations. With proper planning and care, this could even be turned into a suite for testing different signal (buy/sell/keep) generating rules, that compares their results to each other to see which one fares better.

6.2.9 Further Development of the Visual Component

In Section 2.9 several visualization alternatives were reviewed. However, the visual component of the tool could still be further developed. There is some existing literature that can help improve that component. Namely the book "Information Visualization in Data Mining and Knowledge Discovery" [Fay+01], the papers "Visualizing Sequential Patterns for Text Mining (2000)" [Won+00] and "CrystalClear: Active Visualization of Association Rules (2002)" [Ong+02], and the article "Information Visualization and Visual Data Mining" [Kei02] provide additional information about existing alternatives for big data visualization. These literature has already been used as inspiration for the planning of some visualization components. However, future development of this component was considered beyond the scope of the prototype developed for this thesis.

BIBLIOGRAPHY

[Aal+10] W. M. van Aalst, K. M. van Hee, J. M. van Werf, and M. Verdonk. "Auditing 2.0: Using Process Mining to Support Tomorrow's Auditor". In: *Computer* 43.3 (2010), pp. 90–93. ISSN: 0018-9162. DOI: http://doi.ieeecomputersociety.org/10.1109/MC.2010.61.

[AS94] R. Agrawal and R. Srikant. "Fast Algorithms for Mining Association Rules in Large Databases". In: *Proceedings of the 20th International Conference on Very Large Data Bases*. VLDB '94. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1994, pp. 487–499. ISBN: 1-55860-153-8. URL: http://dl.acm.org/citation.cfm?id=645920.672836.

[AF94] J. F. Allen and G. Ferguson. "Actions and Events in Interval Temporal Logic". In: *Journal of Logic and Computation* 4 (1994), pp. 531–579.

[Ayr+02] J. Ayres, J. Flannick, J. Gehrke, and T. Yiu. "Sequential PAttern mining using a bitmap representation". In: *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. KDD '02. Edmonton, Alberta, Canada: ACM, 2002, pp. 429–435. ISBN: 1-58113-567-X. DOI: 10.1145/775047.775109. URL: http://doi.acm.org/10.1145/775047.775109.

[Blo] *Bloomberg*. http://www.bloomberg.com/. Accessed: 2014-09-16.

[Bor+96] S. Borenstein, A. Shepard, and N. B. of Economic Research. *Sticky Prices, Inventories, and Market Power in Wholesale Gasoline Markets*. NBER working paper series no. 5468. National Bureau of Economic Research, 1996. URL: http://books.google.pt/books?id=PUhnmgEACAAJ.

[BC02] D. Byrd and T. Crawford. "Problems of music information retrieval in the real world". In: *Information Processing and Management*. 2002, pp. 249–272.

[Cav07a] L. Cavique. "A network algorithm to discover sequential patterns". In: *Proceedings of the aritficial intelligence 13th Portuguese conference on Progress in artificial intelligence*. EPIA'07. Guimarães, Portugal: Springer-Verlag, 2007, pp. 406–414. ISBN: 3-540-77000-3, 978-3-540-77000-8. URL: http://dl.acm.org/citation.cfm?id=1782254.1782294.

[CC08] L. Cavique and J. Coelho. "Descoberta de padrões sequenciais utilizando árvores orientadas". In: *Revista de Ciências da Computação* 3.3 (2008).

[Cav07b] L. Cavique. "A scalable algorithm for the market basket analysis". In: *Journal of Retailing and Consumer Services* 14.6 (2007). Data Mining Applications in Retailing and Consumer Services, pp. 400 –407. ISSN: 0969-6989. DOI: 10.1016/j.jretconser.2007.02.003. URL: http://www.sciencedirect.com/science/article/pii/S0969698907000148.

[CM13] L. Cavique and N. C. Marques. "Sequential Pattern Mining of Price Interactions". In: *EPIA 2013, 16th Portuguese Conference on Artificial Intelligence, to appear*. Angra do Heroísmo, Açores, Portugal, 2013.

[Che+10] Y.-C. Chen, J.-C. Jiang, W.-C. Peng, and S.-Y. Lee. "An efficient algorithm for mining time interval-based patterns in large database". In: *Proceedings of the 19th ACM international conference on Information and knowledge management*. CIKM '10. Toronto, ON, Canada: ACM, 2010, pp. 49–58. ISBN: 978-1-4503-0099-5. DOI: 10.1145/1871437.1871448. URL: http://doi.acm.org/10.1145/1871437.1871448.

[D3j] *D3JS Main Page*. http://d3js.org/. Accessed: 2013-06-24.

[Das99] S. Dasgupta. "Learning Polytrees". In: *UAI '99: Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence, Stockholm, Sweden, July 30 - August 1, 1999*. 1999, pp. 134–141. URL: http://uai.sis.pitt.edu/displayArticleDetails.jsp?mmnu=1&smnu=2&article_id=162&proceeding_id=15.

[Don+05] B. F. van Dongen, A. K. A. de Medeiros, H. M. W. Verbeek, A. J. M. M. Weijters, and W. M. P. van der Aalst. "The Prom Framework: A New Era in Process Mining Tool Support". In: *Proceedings of the 26th International Conference on Applications and Theory of Petri Nets*. ICATPN'05. Miami: Springer-Verlag, 2005, pp. 444–454. ISBN: 3-540-26301-2, 978-3-540-26301-2. DOI: 10.1007/11494744_25. URL: http://dx.doi.org/10.1007/11494744_25.

[Eia] *EIAU.S Energy Information Administration*. http://www.eia.gov. Accessed: 2014-04-29.

[Fay+01] U. Fayyad, G. G. Grinstein, and A. Wierse. *Information Visualization in Data Mining and Knowledge Discovery*. 1st. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001. ISBN: 1558606890.

[Stl] *Federal Reserve Bank of St. Louis*. http://research.stlouisfed.org/. Accessed: 2014-09-16.

[GH01] J. Gary and G. Handwerk. *Petroleum Refining*. Institut français du pétrole publications. Taylor & Francis, 2001. ISBN: 9780824745172. URL: http://books.google.pt/books?id=eE3_IqDeeosC.

[Gep] *Gephi Main Page*. https://gephi.org. Accessed: 2013-06-18.

[Gra] *GraphViz Main Page*. http://www.graphviz.org/. Accessed: 2013-06-24.

[GQ11] T. Guyet and R. Quiniou. "Extracting temporal patterns from interval-based sequences". In: *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence - Volume Volume Two*. IJCAI'11. Barcelona, Catalonia, Spain: AAAI Press, 2011, pp. 1306–1311. ISBN: 978-1-57735-514-4. DOI: 10.5591/978-1-57735-516-8/IJCAI11-221. URL: http://dx.doi.org/10.5591/978-1-57735-516-8/IJCAI11-221.

[HC] M. Hahsler and S. Chelluboina. *Visualizing Association Rules: Introduction to the R-extension Package arulesViz*.

[Ham+08] S. Hammoudeh, B. T. Ewing, and M. A. Thompson. "Threshold Cointegration Analysis of Crude Oil Benchmarks". In: *The Energy Journal* Volume 29.Number 4 (2008), pp. 79–96. URL: http://EconPapers.repec.org/RePEc:aen:journl:2008v29-04-a04.

[HK06] J. Han and M. Kamber. *Data Mining, Southeast Asia Edition: Concepts and Techniques*. The Morgan Kaufmann Series in Data Management Systems. Elsevier Science, 2006. ISBN: 9780080475585. URL: http://books.google.cz/books?id=AfL0t-YzOrEC.

[Har+10] D. Harmon, B. Stacey, Y. Bar-Yam, and Y. Bar-Yam. *Networks of Economic Market Interdependence and Systemic Risk*. Papers. arXiv.org, 2010. URL: http://EconPapers.repec.org/RePEc:arx:papers:1011.3707.

[IK00] R. G. Ibbotson and P. D. Kaplan. "Does asset allocation policy explain 40, 90, or 100 percent of performance". In: *Financial Analysts Journal* (2000), pp. 26–33.

[Sps] *IBM SPSS*. http://www.ibm.com/software/analytics/spss/. Accessed: 2014-09-22.

[Kei02] D. A. Keim. "Information Visualization and Visual Data Mining". In: *IEEE Transactions on Visualization and Computer Graphics* 8.1 (Jan. 2002), pp. 1–8. ISSN: 1077-2626. DOI: 10.1109/2945.981847. URL: http://dx.doi.org/10.1109/2945.981847.

[Kos+00] N. Kosugi, Y. Nishihara, T. Sakata, M. Yamamuro, and K. Kushima. "A Practical Query-by-humming System for a Large Music Database". In: *Proceedings of the Eighth ACM International Conference on Multimedia*. MULTIMEDIA '00. Marina del Rey, California, USA: ACM, 2000, pp. 333–342. ISBN: 1-58113-198-4. DOI: 10.1145/354384.354520. URL: http://doi.acm.org/10.1145/354384.354520.

[Kur+02] F. Kurth, A. Ribbrock, and M. Clausen. "Efficient Fault Tolerant Search Techniques for Full-Text Audio Retrieval". In: *Audio Engineering Society Convention 112*. 2002. URL: http://www.aes.org/e-lib/browse.cfm?elib=11335.

[MG10] N. C. Marques and C. Gomes. "Implementing an Intelligent Moving Average with a Neural Network". In: *Proceedings of the 2010 Conference on ECAI 2010: 19th European Conference on Artificial Intelligence*. Amsterdam, The Netherlands, The Netherlands: IOS Press, 2010, pp. 1129–1130. ISBN: 978-1-60750-605-8. URL: http://dl.acm.org/citation.cfm?id=1860967.1861239.

[McN+08] P. D. McNicholas, T. B. Murphy, and M. O'Regan. "Standardising the Lift of an Association Rule". In: *Comput. Stat. Data Anal.* 52.10 (June 2008), pp. 4712–4721. ISSN: 0167-9473. DOI: 10.1016/j.csda.2008.03.013. URL: http://dx.doi.org/10.1016/j.csda.2008.03.013.

[Ong+02] K. huat Ong, K. leong Ong, W.-K. Ng, and E.-P. Lim. "CrystalClear: Active Visualization of Association Rules". In: *In ICDM'02 International Workshop on Active Mining AM2002*. Press, 2002.

[Pam+02] E. Pampalk, A. Rauber, and D. Merkl. "Content-based Organization and Visualization of Music Archives". In: *Proceedings of the Tenth ACM International Conference on Multimedia*. MULTIMEDIA '02. Juan-les-Pins, France: ACM, 2002, pp. 570–579. ISBN: 1-58113-620-X. DOI: 10.1145/641007.641121. URL: http://doi.acm.org/10.1145/641007.641121.

[Pap+09] P. Papapetrou, G. Kollios, S. Sclaroff, and D. Gunopulos. "Mining frequent arrangements of temporal intervals". In: *Knowl. Inf. Syst.* 21.2 (Oct. 2009), pp. 133–171. ISSN: 0219-1377. DOI: 10.1007/s10115-009-0196-0. URL: http://dx.doi.org/10.1007/s10115-009-0196-0.

[Pei+01] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu. "PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth". In: *Proceedings of the 17th International Conference on Data Engineering*. ICDE '01. Washington, DC, USA: IEEE Computer Society, 2001, pp. 215–. URL: http://dl.acm.org/citation.cfm?id=876881.879716.

[Por] *Pordata*. http://www.pordata.pt/. Accessed: 2014-09-16.

[RF01] A. Rauber and M. Frühwirth. "Automatically Analyzing and Organizing Music Archives". In: *Proceedings of the 5th European Conference on Research and Advanced Technology for Digital Libraries*. ECDL '01. London, UK, UK: Springer-Verlag, 2001, pp. 402–414. ISBN: 3-540-42537-3. URL: http://dl.acm.org/citation.cfm?id=646634.699932.

[Red99] I. Redbooks. *Intelligent Miner for Data Applications Guide*. IBM redbooks. Vervante, 1999. ISBN: 9780738412764. URL: http://books.google.pt/books?id=9skAAAAACAAJ.

[RA08] A. Rozinat and W. M. P. van der Aalst. "Conformance Checking of Processes Based on Monitoring Real Behavior". In: *Inf. Syst.* 33.1 (Mar. 2008), pp. 64–95. ISSN: 0306-4379. DOI: 10.1016/j.is.2007.07.001. URL: http://dx.doi.org/10.1016/j.is.2007.07.001.

[Sil+05] A. Silberschatz, H. F. Korth, and S. Sudarshan. *Database System Concepts, 5th Edition*. McGraw-Hill Book Company, 2005. ISBN: 978-0-07-295886-7.

[Suv+09] H. Suviolahti et al. "The influence of volatile raw material prices on inventory valuation and product costing". In: (2009).

[Cia] *The World Factbook - CIA*. https://www.cia.gov/library/publications/the-world-factbook/. Accessed: 2014-09-16.

[Typ+05] R. Typke, F. Wiering, and R. C. Veltkamp. "A Survey Of Music Information Retrieval Systems". In: *IN ISMIR*. 2005, pp. 153–160.

[VDA12] W. Van Der Aalst. "Process Mining". In: *Commun. ACM* 55.8 (Aug. 2012), pp. 76–83. ISSN: 0001-0782. DOI: 10.1145/2240236.2240257. URL: http://doi.acm.org/10.1145/2240236.2240257.

[WF03] A. L. chun Wang and T. F. B. F. "An industrial-strength audio search algorithm". In: *Proceedings of the 4 th International Conference on Music Information Retrieval*. 2003.

[Won+00] P. C. Wong, W. Cowley, H. Foote, E. Jurrus, and J. Thomas. "Visualizing Sequential Patterns for Text Mining". In: *Proc. IEEE Information Visualization, 2000*. Society Press, 2000, pp. 105–114.

[Wor] *World Bank Group*. http://www.worldbank.org/. Accessed: 2014-09-16.

[Zak01] M. J. Zaki. "SPADE: An efficient algorithm for mining frequent sequences". In: *Machine Learning*. 2001, pp. 31–60.

APPENDIX A

APPENDIX

A.1 List of products analyzed in the third case study

3M Company, ASICS Corporation, Abbott Laboratories, Agilent Technologies, Air Liquide, America Movil 12, Asian Development Bank 10, Assa Abloy, BL Global Equities B, BNDES 10-17, Ball Corp, Bayer AG, Brazil 05-15, British American Tabacco, CIA Cervecerias Unidas, Cafe de Coral Holdings, Canon, Capital Gestion Multi Bond, Cisco Systems, Coca-Cola Co, DO&CO, DON QUIJOTE CO, Dairy Farm International Holdings, Deutschland 07, Deutschland 09, Deutschland 12-17, EMC Corp, Ecolab Inc, Emerson Electric Company, European Financial Stability Facility 13, Exxon Mobil, FMC Corp, Fanuc, FedEx, Finland 06-17, Finland 08, Flowserve, Groupe Danone, Hasbro, Hitachi, Hoya, IBRD 13, JSR CORP, Japan Tobacco, Johnson & Johnson, Johnson Controls, KANSAI PAINT CO.LTD, KAO, KEYENCE CORP, Kraft Foods Group, Kuraray, Kyocera, LAWSON INC OSAKA, Laboratory Corporation of America Holdings, Linde, Lowe's Companies, MANDOM CORP, Mattel, Mexico 05-15, Mexico 10-17, Microsoft, Murata Manufacturing, National Oilwell Varco, Nederland 12, Novartis, OBIC Co Ltd, Oracle, PETROBRAS 11, Parker-Hannifin, Pemex 09, Pemex Project Funding Master Trust 04-16, PepsiCo, Peru 04-14, Perushahaan Rokok Tjap Gudang Garam, Poland 06-16, Poland 08-18, Poland 10-21, Praxair, Procter & Gamble, Qualcomm, Reckitt Benckiser Group, Roche, Roper Industries, SAP, SES Global, SKF, SMC Corp, Sage Group, Sandvik, Sanofi, Schneider Electric, Sekisui House, Shin-Etsu Chemical, Shire PLC, Sika, South Africa 06-16, Stryker, Swatch Group, Swedish Match, Syngenta, THE MIDDLEBY CORPORATION, Target, Terumo Corp, Tesco 11, Thermo Fisher Scientific Inc, Tiger Brands, Tingyi (Cayman Islands) Holding, Turkey 06-16, Unicharm CORP, Unilever, Viacom B, Walgreen.